

摘要

计算机代数又称符号或代数计算（或数学机械化），它研究用计算机取代人类的大脑去处理各种复杂的符号计算和任何种类数学运算过程，也就是让数学的研究走向机械化的过程。这一研究课题旨在让人类将大量繁琐的计算问题交给计算机去处理。从而使人类从繁琐的计算工作中解放出来，而将自己的聪明才智与创造能力贯注到更有意义的脑力劳动上去。伴随着微分方程在现代科技、工程领域中日益显著的重要作用，与此同时，在实际应用中，大多数微分方程的求解问题相当复杂却又是解决问题的关键，寻找简便可行的计算方法在微分方程模型求解的实际应用和研究中的重要作用勿容置疑。

本课题研究用计算机代数求解微分方程的理论与方法及其机器实现。本文的第三章研究了基于吴特征列方法的双曲正切函数展开法，给出了求解比较复杂的偏微分方程的通用求解过程；在第三章，我们在 Adomian 分解原理的基础上，给出了利用 Adomian 分解原理求解一类奇异初值问题的算法及其 Maple 环境下的程序，解决了用手工过程求解无法实现的求解问题；本文的第四章研究了微分代数方程的幂级数展开法和 Pade 级数近似解，给出了相应的机器求解过程，并给出了 Pade 级数的精确解和近似数值解在同一坐标系下的图形，给出了二者的逼近比较；第五章作出了将符号计算方法和数值计算方法结合起来求解微分方程的研究工作，这是求解比较复杂的偏微分方程的新途径。这一章的意义在于引导我们求解复杂的偏微分方程的新思维和新视角。本文在求解过程中，运用了计算机代数的有关知识并且借助了计算机代数系统中的 Maple 和 Matlab 软件。求解过程相对手工过程而言，具有极大优势。

关键词 计算机代数，微分方程，求解，方法，机器实现

ABSTRACT

Computer Algebra is also called symbolic or algebra computation (or mathematics mechanization). The subject tries to deal with mathematics in a constructive and algorithmic manner so that the reasoning become mechanical, automated and so much as possible as to be intelligence lacking, with the result of lessening the painstaking heavy brain-labor. It is a new development orientation in the field of mathematics and computer to conduct science calculation by computer. For a long time, the mathematicians and computer scientists have dreamed of replacing human brain with computer to conduct symbolic operation and any kind of mathematical processes, leading the mathematics to a mechanizing way. Nowadays, differential equations is playing a more and more important role in modern science and engineering field, and at the same time, in practice, most solution of differential equations are extremely sophisticated but crucial. Seeking for the simple and practicable computation method is of great concern in the practical application and research of the solution of the model of the differential equations.

In this paper, it researches the theories, methods and computation mechanization of solving the differential equations with computer algebra knowledge and computer algebra systems. We research the tanh-method on the base of Wu-characteristic method, and give the interchangeable process of solving the more complicated partial differential equations in the third chapter. Furthermore, we give the algorithm of the initial problems of differential equations which include singularity and its procedure in Maple by the use of the method of Adomian decomposition and accomplish the solution which is unable to achieve by hand-work in the third chapter. We research the Power series and the Pade series approximated solution of differential-algebraic equations in the fourth chapter. It gives the solving process by computer and its figures of exact solution and approximated numerical solution of Pade series under the same coordinate axis and we give the analysis and comparison about asymptotic approximation of them. In the fifth chapter, we give the

method of solving the partial differential equations with the combine use of symbolic and numerical method, which is a new way of solving the extremely complicated partial differential equations. Its significance is to conduct us to seeking for the new idea and new sight of solving the complicated partial differential equations. In the process of solving, we use the computer algebra knowledge and Maple or Maple software of computer systems. By the comparison of solving the differential equations with computer and hand-work, the tremendous advantage is opened out and the great practical value is reflected by the using of computer algebra.

KEY WORDS computer algebra, differential equations, solve, method, computation mechanization

原创性声明

本人声明，所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。论文主要是自己的研究所得，除了已注明的地方外，不包含其他人已经发表或撰写过的研究成果，也不包含为获得中南大学或其他单位的学位或证书而使用过的材料。与我共同工作的同志对本研究所作的贡献，已在论文的致谢语中作了说明。

作者签名： 蒋青松 日期： 2006 年 11 月 25 日

关于学位论文使用授权说明

本人了解中南大学有关保留、使用学位论文的规定，即：学校有权保留学位论文，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以采用复印、缩印或其他手段保存学位论文；学校可根据国家或湖南省有关部门的规定，送交学位论文。对以上规定中的任何一项，本人表示同意，并愿意提供使用。

作者签名： 蒋青松 导师签名： 廖文志 日期： 2006 年 11 月 26 日

第一章 绪 论

1.1 背景知识

计算机代数是一门数学与计算机科学的交叉学科, 又称符号与代数计算(或数学机械化), 它研究如何使用计算机来表示和处理数学概念、符号和知识、进行数学的计算和推理、显示和分析数据与图形等问题。计算机代数自 20 世纪 60 年代就得到了深入的研究, 并不断发展, 发展至今, 这门学科已经相对成熟, 它的理论、方法、软件的应用都已经比较完善。计算机代数系统已广泛应用于与数学计算有关的众多学科, 对现代科研与教学产生了积极影响。它的发展始终与代数计算和软件开发联系在一起, 并受到物理计算的激励, 其主要分支包括计算机代数与分析、几何计算、自动推理与编程等。符号计算强调构造性理论的建立与发展、有效算法的设计与实施、软件系统的研制与开发, 以及它们在科学工程中的应用。符号计算、自动推理与吴文俊院士开创并倡导的数学机械化密切相关。值得一提的是我国学者在符号计算的这些领域都有突出的研究成果, 证明几何定理的吴方法更是我国的独创。

众所周知, 计算机科学和计算机代数系统的快速发展, 对数学研究的观念和方法已经产生了深远的影响, 在数学和计算机科学中的一个新的发展方向就是通过计算机去做科学计算的研究。在很长的一段时间里, 数学和计算机工作者都梦想着用计算机取代人类的大脑去处理各种复杂的符号计算和任何种类数学运算过程, 也就是让数学的研究走向机械化的过程。从而计算机本身更加智能化, 这就直接导致了数学的机械化这门课题的诞生与发展。数学机械化这门研究课题, 就是极力地将各类数学问题付诸于建设性的、可行的、算法化的方式来处理, 这样便使得数学问题的机械化和自动化成为现实, 从而人类可以将大量繁琐的计算问题交给计算机去处理。数学机械化是人类科学历史上的一次重大的改革, 它将人类从繁复的计算工作中解放出来, 使人类能将自己的聪明才智与创造能力贯注到更有意义的脑力劳动上去。

1.2 课题来源及研究意义

微分方程是伴随着微积分学一起发展起来的学科,如今它已广泛应用于物理,化学,生物,经济,动力系统,工程力学等领域,研究微分方程的重要性不言而喻。本文研究用计算机代数去求解微分方程的理论与方法。这一研究课题源于微分方程应用的广泛性和日益显著的重要性,很多实际问题的解决都取决于相应的微分方程模型的求解,从而使得微分方程的求解问题在实际应用中显得尤为重要。一方面,微分方程模型的求解在实际问题的研究和应用中的重要性日益显著,而与此同时微分方程的求解问题历来又成为困扰人类解决实际问题的一大障碍,这一切都是因为很多微分方程的求解过程是很繁琐和枯燥的,人们迫切需要找到微分方程的简便可行的求解方法。在这方面的研究领域中,大量的科学工作者投入其中,并作出了相应的贡献,在此不一一列举。但已有的研究成果很不完善,很多实际问题仍然无法得以解决。伴随着计算机科学和计算机代数系统的发展,数学研究的观念和方法已经发生了一场空前的变革,过去很多只能用手工处理的复杂的计算问题,如今可以通过利用计算机非常便利快捷的解决。伴随着计算机科学事业的飞速发展,过去的一些利用手工过程计算起来非常繁琐枯燥的计算问题,如今交给计算机去处理可以非常简便快速的解决。因此对于过去用手工过程计算,非常烦琐复杂的微分方程模型的求解问题,如今借助于计算机通过编制算法程序或利用计算机代数系统得以便捷的解决成为可能。尽管计算机代数自20世纪60年代就得到了深入的研究,也有大量的科学工作者投身于数学的机械化的研究工作中(这方面最突出的贡献便是我国吴文俊院士开创并倡导的数学机械化),但数学的机械化是个需要不断发展和成长的研究领域,过去的科学研究者所作的工作并不完善,需要不断的补充和完善。在此大环境下,我们研究了利用计算机代数去求解微分方程的问题,通过利用计算机代数求解微分方程和手工过程求解微分方程的比较,我们可以发现利用计算机代数求解微分方程的极大优势和巨大潜力。用计算机代数求解微分方程的方法使得过去无法求解的复杂的微分方程模型,如今借助计算机通过编制算法程序或借助已有计算机代数软件得以实现。过去利用手工过程求解非常繁琐枯燥的微分方程模型如今借助计算机代数去求解变得非常便捷。用计算机代数去研究微分方程的求解问题,使得我们从过去枯燥烦琐的手工计算中得以解放,很多过去无法想象的微分方程模型的求解问题如今得以圆满解决成为可能。这使我们可以将更多的智慧和精力投诸于更有意义的工作中去。由此可见,这一研究课题迎合时代发展的脉搏,具有重大的现实意义和实际价值。

1.3 本文的结构、思想方法

本文的研究课题为计算机代数求解微分方程的方法研究及其机器实现。全文正文部分共分五章来完成课题的论述。第一章为绪论, 主要介绍课题的背景知识、来源、研究意义以及全文的结构和研究成果。第二章为预备知识, 主要介绍了计算机代数在求解微分方程中运用到的相关知识和理论, 这些知识分别包含微分方程的知识和计算机代数的相关知识。有关微分方程的知识主要介绍微分方程求解的理论依据——Banach 空间中解的存在唯一性以及解对初值的连续依赖性。有关计算机代数部分的知识主要介绍吴—Ritt 特征列方法。从第三章到第五章为论文的主体部分。第三章介绍计算机代数求解微分方程的符号计算方法, 主要内容包括用计算机代数知识吴—Ritt 特征列方法来处理求解微分方程中遇到的难点, 给出了基于吴—Ritt 特征列方法求解微分方程的通用求解方法(双曲正切函数展开法)的求解过程以及用 Adomian 分解方法来求解一类特殊的奇异初值问题的理论、算法、程序和算例; 在第三章中还简要介绍了求解微分方程的比较经典的其他方法——用对称变换方法求解微分方程的机器实现方法。第四章为用计算机代数求解微分方程的数值计算方法, 主要内容包含用幂级数展开法和 Pade 级数法来求解微分方程的理论、算法、程序和算例; 并给出了相应的 Pade 级数近似解和精确解的函数图形, 作出了相应的比较。全文的第五章为用符号—数值混合计算方法来求解偏微分方程。其主要思想方法是将计算机代数知识中经典的吴—Ritt 特征列方法和 Banach 空间中改进的 Newton 方法相结合来求解比较复杂的偏微分方程。其思想方法是首先将一个比较复杂的偏微分方程转化为代数方程组(比如利用双曲正切函数展开法), 然后对代数方程组中的代数多项式组, 利用吴—Ritt 特征列方法(又称作吴—Ritt 整序原理)加以整理变换, 求得此代数多项式组的特征列多项式组, 这时原偏微分方程的解就等价于新的特征列多项式方程组的解。如果待求解的特征列方程组形式比较简单(譬如所含变量较少的情形), 那么我们就可以直接求得其精确解, 如果待求解的特征列方程组为含多变量多参数的复杂代数方程组, 那么我们就不能求得其精确解。在实际应用中, 在精度允许的范围内, 求得难于求解精确解的偏微分方程的近似数值解已经足够。基于此, 符号—数值混合计算求解偏微分方程的方法就很值得我们去探索和研究。在已求得原偏微分方程的等价特征列多项式方程组的基础上, 我们再引入改进的 Newton 迭代法(在 Banach 空间中, 对经典 Newton 迭代法加以改进, 从而经典 Newton 迭代法中只对连续实值函数成立的情形拓展到对抽象函数也成立), 利用 Banach 空间中改进的 Newton 迭代法可以求得等价的待求解的特征多项式方

程组的近似数值解,通过对初值和参数的调整,在计算机上运行,就可以求得满足精度要求的近似数值解,(理论上,精度可以依人为设想而具任意精确度,但实际中限于计算机内存只能具备有限精确度,但在一般情况下,这已经符合要求)。从而求得原偏微分方程满足精度要求的近似数值解。

1.4 本课题的研究成果和展望

本课题研究用计算机代数求解微分方程的理论与方法及其机器实现。本文的第三章研究了基于吴特征列方法的双曲正切函数展开法,给出了求解比较复杂的偏微分方程的通用求解过程;在第三章,我们在 Adomian 分解原理的基础上,给出了利用 Adomian 分解原理求解一类奇异初值问题的算法及其 Maple 环境下的程序,解决了用手工过程求解无法实现的求解问题;本文的第四章研究了微分代数方程的幂级数展开法和 Pade 级数近似解,给出了相应的机器求解过程,并给出了 Pade 级数的精确解和近似数值解在同一坐标系下的图形,给出了二者的逼近比较;第五章作出了将符号计算方法(吴—Ritt 特征列方法)和数值计算方法(Banach 空间中改进的 Newton 迭代法)结合起来求解微分方程的研究工作,这是求解比较复杂的偏微分方程的新途径。我们还可以尝试将其他符号方法和数值方法相结合来求解微分方程的途径。这一章的意义在于引导我们求解复杂的偏微分方程的新思维和新视角。这一研究方向值得我们去探索。本文在求解过程中,运用了计算机代数的有关知识并且借助了计算机代数系统中的 Maple 和 Matlab 软件。求解过程相对手工过程而言,具有极大优势。本人在文中所作的研究工作只是一个初步探索,在计算机代数求解微分方程这一领域中有大量的工作等着我们去探索和研究。希望有更多的人关注这一领域的研究工作,为数学机械化事业作出应用的贡献。

第二章 预备知识

2.1 抽象函数的积分与微分

本节简要介绍一下抽象函数的积分与微分, 本节的主要内容取之于文献[1], 有关引理和定理的证明过程均可在文中找到, 从略。

设 E 是 *Banach* 空间。自变量为实数 $t \in [a, b]$, 而函数值为 $x(t) \in E$ 的函数称为抽象函数或向量函数, 记为 $x(t): [a, b] \rightarrow E$ 。

定义 2.1.1 设 $x(t): [a, b] \rightarrow E, t \in [a, b]$. 如果对任意 $\varepsilon > 0$, 存在 $\delta = \delta(\varepsilon, t) > 0$ 使得:

$$\|x(t + \Delta t) - x(t)\| < \delta, \text{ 其中 } |\Delta t| < \delta,$$

则称 $x(t)$ 在 t 点连续。如果 $x(t)$ 在 $[a, b]$ 上每一点连续, 则称 $x(t)$ 在 $[a, b]$ 上连续。

抽象连续函数有如下基本性质:

引理 2.1.2 如果抽象函数 $x(t)$ 在 $[a, b]$ 上连续, 则 $x(t)$ 在 $[a, b]$ 上一致连续, 则对任意的 $\varepsilon > 0$, 存在 $\delta = \delta(\varepsilon, t) > 0$ 使得

$$\|x(t) - x(t')\| < \varepsilon, \text{ 当 } t, t' \in [a, b], |t - t'| < \delta.$$

定义 2.1.3 设 $x(t): [a, b] \rightarrow E$ 是一个抽象函数, 如果存在 $z \in E$, 使得: 对 $[a, b]$ 的任一分法

$$T: a = t_0 < t_1 < \cdots < t_{i-1} < t_i < \cdots < t_n = b \quad (2.1.1)$$

及任意 $\xi_i \in [t_{i-1}, t_i] (i = 1, 2, \dots, n)$ 所作的积分和

$$\sigma = \sum_{i=1}^n x(\xi_i) \Delta t_i, \Delta t_i = t_i - t_{i-1}$$

满足

$$\text{当 } d(T) = \max_{1 \leq i \leq n} \Delta t_i \rightarrow 0 \text{ 时有 } \|\sigma - z\| \rightarrow 0,$$

则称 $x(t)$ 在 $[a, b]$ 上 Riemann 可积, z 称为 $x(t)$ 在 $[a, b]$ 上的 Riemann 积分 (简称为积分), 记为

$$z = \int_a^b x(t) dt = \lim_{d(t) \rightarrow 0} \sigma,$$

$$\text{即 } \int_a^b x(t) dt = \lim_{d(T) \rightarrow 0} \sum_{i=1}^n x(\xi_i) \Delta t_i. \quad (2.1.2)$$

定理 2.1.4 如果抽象函数 $x(t)$ 在 $[a, b]$ 上连续, 则 $x(t)$ 在 $[a, b]$ 上可积。

引理 2.1.5 如果抽象函数 $x(t): [a, b] \rightarrow E$ 在 $[a, b]$ 上连续, 则

$$\left\| \int_a^b x(t) dt \right\| \leq \int_a^b \|x(t)\| dt. \quad (2.1.3)$$

定义 2.1.6 设 $x(t): [a, b] \rightarrow E, t \in [a, b]$. 如果存在 $y \in E$ 使得

$$\lim_{h \rightarrow 0} \left\| \frac{x(t+h) - x(t)}{h} - y \right\| = 0,$$

则称 $x(t)$ 在 t 处可微. 称 y 为 $x(t)$ 在 t 处的导数, 记为 $y = x'(t)$, 即

$$x'(t) = \lim_{h \rightarrow 0} \frac{x(t+h) - x(t)}{h}. \quad (2.1.4)$$

如果 $x(t)$ 在 $[a, b]$ 中的每一点均可微 (在 a 处右可微, 在 b 处左可微), 则称 $x(t)$ 在 $[a, b]$ 上可微. 导数 $x'(t): [a, b] \rightarrow E$ 也是一个抽象函数。

引理 2.1.7 如果 $x(t): [a, b] \rightarrow E$ 在 t 处可微, 则 $x(t)$ 在 t 处连续。

定理 2.1.8 设 $x(t): [a, b] \rightarrow E$ 在 $[a, b]$ 上连续, 令

$$y(t) = \int_a^t x(s) ds, t \in [a, b],$$

则 $y'(t)$ 在 $[a, b]$ 上可微, 且

$$y'(t) = x(t), \forall t \in [a, b].$$

2.2 Banach 空间中的常微分方程初值问题

本节简要介绍一下 Banach 空间中的常微分方程初值问题, 本节的主要内容取之于文献[1], 有关定理和命题的证明过程均可在文中找到, 从略。

设 E 是 Banach 空间, 算子 $f: E \rightarrow E$. 我们将对 Banach 空间中的常微分方程

初值问题

$$\frac{dx(t)}{dt} = f(x(t)), x(t_0) = u, u \in E \quad (2.2.1)$$

建立解的存在唯一性定理及解对初值的连续依赖性。为此，我们要用到下面的压缩映像原理。

定理 2.2.1 设 X 是 Banach 空间， Σ 是 X 中的闭集，算子 $A: X \rightarrow X$ 在 Σ 上是压缩的，即

$$Ax \in \Sigma, \forall x \in \Sigma, \quad (2.2.2)$$

且存在正数 $\theta < 1$ 使得：

$$\|Ax - Ay\| \leq \theta \|x - y\|, \forall x, y \in \Sigma, \quad (2.2.3)$$

则存在唯一的 $x \in \Sigma$ 使得

$$Ax = x. \quad (2.2.4)$$

满足等式 (2.2.4) 的 x 称为算子的不动点。

定理 2.2.2 设 E 是 Banach 空间，算子 $f: E \rightarrow E$ 满足条件：存在常数 $M > 0$ ，使得

$$\|f(x)\| \leq M, \forall x \in E. \quad (2.2.5)$$

再设算子 f 是局部 Lipschitz 连续的，

即对任意元素 $u \in E$ ，存在常数 $R = R(u) > 0, K = K(u) > 0$ ，使得

$$\|f(x) - f(y)\| \leq K \|x - y\|, \forall x, y \in B_R(u). \quad (2.2.6)$$

那么：

(i) 对任意满足：

$$0 < \delta < \min\left(\frac{R}{M}, \frac{1}{K}\right) \quad (2.2.7)$$

的 δ ，常微分方程初值问题 (2.2.1) 在 $[t_0 - \delta, t_0 + \delta]$ 上有唯一的解 $x(t)$ ，它满足

$$\|x(t) - u\| \leq R, \forall t \in [t_0 - \delta, t_0 + \delta]. \quad (2.2.8)$$

(ii) 常微分方程初值问题 (2.2.1) 在 $(-\infty, +\infty)$ 中有唯一的解 $x(t) = x(t, t_0, u)$ 。

当 $t_0 = 0$ 时，记 $x(t, 0, u) = x(t, u)$ 。

(iii) 对任意的 $c > t_0$ (或 $c < t_0$)，初值问题 (2.2.1) 的解 $x(t, t_0, u)$ 关于 $t \in [t_0, c]$

(或 $t \in [c, t_0]$) 是初值 u 的一致连续函数, 即对任意的 $\varepsilon > 0$, 存在 $r = r(\varepsilon) = r(\varepsilon, t_0, c, u)$ 使得当 $v \in E, \|v - u\| < r$ 时, 有:

$$\|x(t, t_0, v) - x(t, t_0, u)\| < \varepsilon, \forall t \in [t_0, c] (\text{或 } t \in [c, t_0]). \quad (2.2.9)$$

命题 (P) 对任意的 $t^* \in (-\infty, +\infty)$, 存在 $r^* > 0, \delta^* > 0$, 使得对任意的 $\tau \in I^* = [t^* - \delta^*, t^* + \delta^*]$ 及相应的 $x_\tau = x(\tau, t_0, u)$, 如果 $\|v - x_\tau\| \leq r^*$, 则在 I^* 上存在 $x(t, \tau, v)$ 满足

$$\|x(t, \tau, v) - x(t, t_0, u)\| \leq 2\|v - x_\tau\|, \forall t \in I^*.$$

2.3 计算机代数相关预备知识: 特征列方法

本节主要介绍计算机代数中的特征列方法, 本节的主要内容取之于文献[2]和[3], 读者可以从文献[2]和[3]中找到更详尽的讨论, 相关定理和引理的证明在此从略。

2.3.1 三角列与特征列

设 K 是一特征为零的可计算数域。我们将变元 x 排成固定的次序 $x_1 < x_2 < \dots < x_p$ 。对任意多项式 $P \in K[x] \setminus K$, 我们称使得 $\deg(P, x_p) > 0$ 的最大下标 p 为 P 的类, 记为 $cls(P)$ 。非零常数的类定义为 0。设 $p = cls(P) > 0$, 称 x_p 为多项式 P 的导元, 记为 $lv(P)$, $\deg(P, x_p)$ 为 P 的导次数, 记为 $ldeg(P)$, 而 $lc(P, x_p)$ 为 P 的初式, 记为 $ini(P)$ 。

定义 2.3.1.1 $K[x]$ 中非常数多项式组成的有限非空有序集合

$$T = [T_1, T_2, \dots, T_r]$$

称为三角列或非矛盾拟升列, 如果 $cls(T_1) < cls(T_2) < \dots < cls(T_r)$,

可将任意三角列写成如下形式

$$T = \begin{bmatrix} T_1(x_1, \dots, x_n), \\ T_2(x_1, \dots, x_n, \dots, x_{n_2}), \\ \dots\dots\dots \\ T_r(x_1, \dots, x_n, \dots, x_{n_2}, \dots, x_{n_r}) \end{bmatrix}, \quad (2.3.1)$$

这里

$$\begin{aligned} 0 < p_1 < p_2 < \dots < p_r \leq n, \\ p_i = \text{cls}(T_i), x_{n_i} = \text{lv}(T_i), i = 1, \dots, r. \end{aligned}$$

设 T 为 (2.3.1) 式所示的三角列, 而 P 为任一多项式. 如果 $\text{deg}(P, x_{n_i}) < l \text{deg}(T_i)$, 对所有的 i 成立, 则称 P 对 TS 是约化的.

将多项式 $R = \text{prem}(\dots \text{prem}(P, T_r, x_{n_r}), \dots, T_1, x_{n_1})$

简记为 $\text{prem}(P, TS)$, 并称其为 P 对 TS 的伪余式. 有如下伪余公式

$$I_1^{q_1} \dots I_r^{q_r} P = \sum_{i=1}^r Q_i T_i + R, \quad (2.3.2)$$

这里 q_i 是非负整数, 而

$$I_i = \text{ini}(T_i), Q_i \in K[x], i = 1, \dots, r.$$

明显地, 在 P 对 TS 是约化的情形有 $\text{prem}(P, TS) = P$. 对任意多项式组 PS , $\text{prem}(PS, TS)$ 代表 $\{\text{prem}(P, TS) | P \in PS\}$.

引理 2.3.1.2 对 $K[x]$ 中的任意三角列 TS 与多项式 P , 如果 $\text{prem}(P, TS) = 0$, 则 $\text{Zero}(TS / \text{ini}(TS)) \subset \text{Zero}(P)$.

定义 2.3.1.3 三角列 $T = [T_1, T_2, \dots, T_r] \subset K[x]$ 称为非矛盾升列, 如果对所有的 $2 \leq i \leq r$, T_i 对 $[T_1, T_2, \dots, T_{i-1}]$ 都是约化的. 每个单个非零常数构成的集合都称为矛盾升列.

定义 2.3.1.4 升列 CS 称为非空多项式组 $PS \subset K[x]$ 的特征列, 如果

$$CS \subset \langle PS \rangle, \text{prem}(PS, CS) = \{0\}.$$

命题 2.3.1.5 设 $CS = [C_1, \dots, C_r]$ 为多项式方程组 $PS \subset K[x]$ 的特征列, 且命

$$I_i = \text{ini}(C_i), PS_i = PS \cup \{I_i\}, i = 1, \dots, r,$$

$$IS = \text{ini}(CS) = \{I_1, \dots, I_r\}.$$

则

$$\text{Zero}(CS / IS) \subset \text{Zero}(PS) \subset \text{Zero}(CS), \tag{2.3.3}$$

$$\text{Zero}(PS) = \text{Zero}(CS / IS) \cup \bigcup_{i=1}^r \text{Zero}(PS_i) \tag{2.3.4}$$

在 K 以及 K 的任意扩域中成立。

2.3.2 吴—Ritt 算法

定义 2.3.2.1 称 $K[x]$ 中的非零多项式 F 比 G 有较低的秩, 记作 $F < G$ 或 $F > G$,

如果 $\text{cls}(F) < \text{cls}(G)$ 或者 $\text{cls}(F) = \text{cls}(G) > 0$, 且 $l \deg(F) < l \deg(G)$. 这时, 也称 G 比 F 有较高的秩。

如果 $F < G$ 和 $F > G$ 都不成立, 则称 F 和 G 具有相同的秩, 记作 $F \sim G$.

定义 2.3.2.2 对任意三角列

$$T = [T_1, T_2, \dots, T_r], \quad T' = [T'_1, T'_2, \dots, T'_{r'}].$$

如果存在 $j \leq \min(r, r')$ 使得

$$T_1 \sim T'_1, \dots, T_{j-1} \sim T'_{j-1}, \text{ 而 } T_j > T'_j;$$

或者 $r' > r$, 且 $T_1 \sim T'_1, \dots, T_r \sim T'_r$, 则称 T 比 T' 有较高的秩, 记作 $T > T'$ 或 $T' < T$. 此时, 也称 T' 比 T 有较低的秩。如果 $T < T'$ 和 $T' < T$ 都不成立, 则称 T 和 T' 具有相同的秩, 记作 $T \sim T'$. 这时有

$$r' = r \text{ 且 } T_1 \sim T'_1, \dots, T_r \sim T'_r.$$

考虑任意非空多项式组 PS . 设 Φ 为所有包含于 PS 的升列构成的集合。由于每个多项式都构成升列, 所以 $\Phi \neq \emptyset$. 称 Φ 的任意极小升列为 PS 的基列, 这样的基列存在, 并按如下步骤进行:

从 $PS = FS_1$ 开始, 我们在其中选取一个秩最低的多项式, 譬如 B_1 . 若

$\text{cls}(B_1) = 0$, 则 $[B_1]$ 是 PS 的一基列。否则, 命

$$FS_2 = \{F \in FS_1 \setminus \{B_1\} \mid F \text{ 对 } B_1 \text{ 是约化的}\}.$$

如果 $FS_2 = \emptyset$, 那么 $[B_1]$ 是 $FS_1 = PS$ 的一基列, 由 B_1 的选取可知, FS_2 中的所有多项式都比 B_1 有较高的秩。现设 B_2 为 FS_2 中秩最低的多项式, 且命

$$FS_3 = \{F \in FS_2 \setminus \{B_2\} \mid F \text{ 对 } B_2 \text{ 是约化的}\}.$$

若 $FS_3 = \emptyset$, 则 $[B_1, B_2]$ 为 PS 的一基列。否则, 从 FS_3 中选取一秩最低的多项式 B_3 , 并按上述方式继续进行。由于

$$cls(B_1) < cls(B_2) < cls(B_3) < \dots \leq n$$

这一程序必在有限步内停止, 而最终获得 PS 的一基列。

下面算法更确切的给出上述选择过程。

算法 BasSet: $BS := BasSet(P)$. 任给非空多项式组 $PS \subset K[x]$, 本算法计算 PS 的基列 BS .

B1. 命 $FS := PS, BS := \emptyset$.

B2. 重复下列步骤直至 $FS := \emptyset$.

B2. 1. 从 FS 中选取一秩最低的多项式 B .

B2. 2. 命 $BS := BS \cup \{B\}$

B2. 3. 如果 $cls(B) = 0$, 则命 $FS := \emptyset$; 否则命

$$FS := \{F \in FS \setminus \{B\} \mid F \text{ 对 } B \text{ 是约化的}\}.$$

下面我们描述吴—Ritt 的特征列的算法, 该算法指出了如何从任意多项式求得其特征列。

算法 CharSet: $CS := CharSet(PS)$. 任给非空多项式组 $PS \subset K[x]$, 本算法计算 PS 的特征列 CS .

C1. 命 $FS := PS, RS := PS$.

C2. 重复下列步骤直至 $RS := \emptyset$:

C2. 1. 计算 $CS := BasSet(FS)$.

C2. 2 如果 CS 是矛盾列, 则命 $RS := \emptyset$; 否则计算

$$RS := \text{prem}(FS \setminus CS, CS) \setminus \{0\},$$

且命 $FS := FS \cup RS$.

引理 2. 3. 2. 3 设 $P \subset K[x]$ 是以

$$BS = [B_1, B_2, \dots, B_r]$$

为基列的非空多项式组, 这里 $cls(B_i) > 0$. 如果 B 是一对 BS 约化的非零多项式,

那么 $PS \cup \{B\}$ 有一基列, 其秩比 BS 的秩低。

可将算法 CharSet 图示如下:

$$\begin{aligned}
 PS &= FS_1 \subset FS_2 \subset \dots \subset FS_m \\
 &\quad \cup \quad \cup \quad \cup \\
 BS_1 \quad BS_2 \quad \dots \quad BS_m &= CS \\
 RS_1 \quad RS_2 \quad \dots \quad RS_m &= \emptyset
 \end{aligned}$$

其中

$$\begin{aligned}
 RS_i &= \text{prem}(FS_i \setminus BS_i, BS_i) \setminus \{0\}, \\
 FS_{i+1} &= FS_i \cup RS_i,
 \end{aligned}$$

且 BS_i 是 FS_i 的基列。

整序原理: 设 CS 为多项式 PS 的一个特征列, 则

$$\left. \begin{aligned}
 \text{Zero}(CS/IP) &\subset \text{Zero}(PS) \subset \text{Zero}(CS), \\
 \text{Zero}(PS/IP) &= \text{Zero}(CS/IP), \\
 \text{Zero}(PS) &= \text{Zero}(CS/IP) + \bigcup_i \text{Zero}(PS + \{I_i\}).
 \end{aligned} \right\}$$

此式中 I_i 为 CS 中多项式 C_i 的初式, IP 是 CS 的初式积。此外, CS 的序不会高于 PS 的序。

2.3.3 多项式组的零点分解

定理 2.3.3.1 (零点分解定理 I) 存在一个算法, 使得对于给定的多项式集 PS , 可在有限步内得到有限个升列 CS_j , 使得

$$\text{Zero}(PS) = \bigcup_j \text{Zero}(CS_j / IP_j), \tag{I}$$

$$\text{Remdr}(PS / CS_j) = \{0\}. \tag{2.3.5}$$

(I) 中的每个 IP_j 为相应升列 CS_j 的初式积。此外, (2.3.5) 中的每个 CS_j 的序都不高于 PS 的序。

定理 2.3.3.2 (零点分解定理 II) 存在一个算法, 使得对于给定的多项式集 PS , 可在有限步内计算出有限多个升列 CS_k , 满足

$$\text{Zero}(PS) = \bigcup_k \text{Zero}(CS_k / ISP_k), \tag{II}$$

$$\text{Remdr}(PS / CS_k) = \{0\}. \tag{2.3.6}$$

(II) 中的每个 ISP_k 是相应的升列 CS_k 的初式隔离子积。此外, (2.3.6) 中的每个 CS_k 的序不高于 PS 的序。

定义 2.3.3.3 上述定理 2.3.3.1 和定理 2.3.3.2 中给出的升列 CS_j 组成的集合称

作 PS 的一个特征序列。

2.3.4 三角列的性质

引理 2.3.4.1 设 TS 为 $K[z]$ 中的不可约三角列, 而 ξ 为 TS 的一般零点, 那么对任

$$\text{任意多项式 } P \in K[z], \text{ prem}(P, TS) = 0 \Leftrightarrow P(\xi) = 0.$$

命题 2.3.4.2. 对 $K[x]$ 中的任意不可约三角列 TS 与多项式 P ,

$$\text{prem}(P, TS) = 0 \Leftrightarrow \text{Zero}(TS / \text{ini}(TS)) \subset \text{Zero}(P)$$

在 K 的扩域中成立。

推论 2.3.4.3 设 $\psi = \{CS_1, \dots, CS_e\}$ 为 $K[x]$ 中的任一多项式组 PS 的不可约特征序列, 那么在 K 的扩域中

$$\begin{aligned} \text{Zero}(PS) = \emptyset &\Leftrightarrow \psi = \emptyset, \\ \text{Zero}(PS) \subset \text{Zero}(P) &\Leftrightarrow \text{prem}(P, CS_i) = 0, \forall i(1 \leq i \leq e). \end{aligned}$$

引理 2.3.4.4 设 $TS = [T_1, \dots, T_r]$ 为 $K[x]$ 中的三角列, P 为任一多项式, 而 $R = \text{res}(P, TS)$, 那么在 $K[x]$ 中可求得多项式 Q 和 Q_1, \dots, Q_r , 使得

$$QP = Q_1T_1 + \dots + Q_rT_r + R$$

命题 2.3.4.5 设 $TS = [T_1, \dots, T_r]$, 为 $K[z]$ 中的正则列, 那么对任意 $1 \leq i \leq r-1$ 以及 $[T_1, \dots, T_i]$ 的任意正则零点 ξ_i 都有

$$\text{ini}(T_{i+1})(\xi_i) \neq 0.$$

定理 2.3.4.6 对 $K[x]$ 中任意正则列 TS 与多项式 P , $\text{Zero}(TS / \text{ini}(TS)) \subset \text{Zero}(P)$

当且仅当存在整数 $d > 0$, 使得 $\text{prem}(p^d, TS) = 0$.

定理 2.3.4.7 设 PS 为 $K[x]$ 中的任一多项式组, TS_1, \dots, TS_e 为 $K[x]$ 中的不可约三角列, 而 US_1, \dots, US_e 为 $K[x]$ 中的多项式组, 使得

$$\begin{cases} \text{Zero}(PS) = \bigcup_{i=1}^e \text{Zero}(TS_i / \text{ini}(TS_i) \cup US_i), \\ \text{prem}(U, TS_i) \neq 0, \forall U \in US_i, 1 \leq i \leq e. \end{cases}$$

成立。那么

$$\text{prem}(PS, TS) = \{0\}, 1 < i < e,$$

$$\text{Zero}(PS) = \bigcup_{i=1}^e \text{Zero}(TS_i / \text{ini}(TS)).$$

定义 2.3.4.8 $K[x]$ 中的正则列 $TS = [T_1, \dots, T_r]$ 称为简单列, 如果对所有 $1 \leq i \leq r-1$ 以及 $[T_1, \dots, T_i]$ 的任意正则零点 ξ_i , 多项式

$$T_{i+1}(\xi_i, y_{i+1}) \in K(\xi_i)[y_{i+1}]$$

在 $K(\xi_i)$ 上都无平方因子。

第三章 符号计算方法

符号计算方法求解(偏)微分方程的方法有很多种。本文主要介绍两种方法,即用基于特征列方法的双曲正切函数展开法求解微分方程和用 Adomian 分解方法求解奇异初值问题。最后我们简要介绍一下在计算机代数系统 Maple 环境下用对称变换方法求解微分方程的机器实现方法。下面我们先介绍孤立波解的双曲正切函数展开法。

3.0 双曲正切函数展开法

双曲正切函数展开法是建立在多数非线性发展方程的孤立波解都具有双曲函数形式的基础上,它的本质在于对所求非线性发展方程的解作先验的假设,即孤立波解是一种局部化解,它可以表示为双曲正切函数的多项式,这样就将非线性发展方程孤立波解的求解问题转化为非线性代数方程的求解问题。在数学上,非线性代数方程组的求解是一个非常困难的问题。吴文俊消元法的建立为非线性代数方程组的求解奠定了理论基础,提供了有效算法。李志斌、张善卿于 1997 年(见文献[6])提出利用吴消元法在符号计算系统上推导非线性发展方程的孤立波解,使得双曲正切函数展开法能够应用于更为复杂或一般的非线性发展方程。为简单起见,下面我们以一个未知函数,两个自变量的方程为例介绍双曲正切函数展开法。对多个自变量以及方程组的情形,求解步骤完全类似。

考虑 1+1 维非线性演化方程:

$$F(u, u_t, u_x, u_{xx}, \dots) = 0 \quad (3.0.1)$$

其中 F 是变元 $u, u_t, u_x, u_{xx}, \dots$ 的多项式,方程 (3.0.1) 描述孤立波 $u(x, t)$ 的动态演化过程。对给定的 F , 如果 (3.0.1) 有双曲正切函数多项式形式的孤立波解 $u(x, t)$, 则根据以下步骤必可求得这些解。

(1) 孤立波是一种特殊的行波, 因此首先对方程 (3.0.1) 施行行波变换

$$u = u(\xi), \xi = k(x - ct) + \xi_0 \quad (3.0.2)$$

其中 k (波数) 和 c (波速) 为待定常数, 而 ξ_0 为任意常数, 通常可取为 0。通过导数代换

$$\frac{\partial}{\partial t} \rightarrow -ck \frac{d}{d\xi}, \frac{\partial}{\partial x} \rightarrow k \frac{d}{d\xi}$$

可将方程 (3.0.1) 化为变量 ξ 的常微分方程

$$F(u, u', u'', \dots) = 0 \quad (3.0.3)$$

(2) 假设方程 (3.0.3) 具有双曲正切函数多项式形式的解, 即:

$$u(\xi) = \sum_{i=0}^m a_i T^i, T = \tanh(\xi) \quad (3.0.4)$$

其中系数 $a_i (i=0, \dots, m)$ 为待定参数。由于双曲正切函数 T 满足关系

$$T' = 1 - T^2$$

故 $u(\xi)$ 的导数依然是 T 的多项式。若以 $O(u(\xi))$ 记 $u(\xi)$ 关于 T 的多项式的最高幂次, 则 $d^p u / d\xi^p$ 的最高幂次为

$$O\left(\frac{d^p u}{d\xi^p}\right) = m + p, p = 1, 2, \dots$$

而 $u^q d^p u / d\xi^p$ 的最高幂次为

$$O\left(u^q \frac{d^p u}{d\xi^p}\right) = (q+1)m + p, q = 0, 1, \dots, p = 1, 2, \dots$$

将 (3.0.4) 代入方程 (3.0.3), 并平衡所得方程中线性最高阶导数项与最高阶非线性项的幂次, 可以确定参数 m , 称 m 为孤立波解的阶数。

(3) 将阶数确定的 (3.0.4) 代入方程 (3.0.3), 合并 T 的同次幂系数并令其为 0, 即得关于待定参数 $a_i (i=0, \dots, m), k, c$ 的非线性代数方程组。

(4) 利用吴消元法求解非线性代数方程组, 确定待定参数 $a_i (i=0, \dots, m)$ 和 k, c 。然后返回原来的变量, 最终可以给出方程 (3.0.1) 的孤立波解

$$u(x, t) = \sum_{i=0}^m a_i \tanh^i[k(x - ct)]. \quad (3.0.5)$$

在正常情况下 m 应为正整数, 于是由上面的算法可以求出方程 (3.0.1) 的闭合解析解 (3.0.5)。然而在有些情况下, 平衡方程 (3.0.3) 时可能出现 m 为分数或负整数的情形。在这种情况下可引入变换

$$u(\xi) = v(\xi)^{\frac{\text{sgn}(m)}{d}}$$

将原方程化为 $v(\xi)$ 的方程, 这里 d 表示 m 的分母, $\text{sgn}(m)$ 表示取 m 的符号 (若 m

为正整数, 则 $\text{sgn}(m)=1, d=1$ 。这样, $v(T)$ 的阶为 m 的分子 (一定是正整数)。
对变换后的方程应用上述方法求解, 此时 $v(\xi)$ 的阶为 m 的分子 (正整数)。

3.1 用特征列方法解决微分方程求解过程中的难题

特征列的概念是 J.F. Ritt 在其有关微分代数的工作中对 (微分) 多项式理想引进的, 但 Ritt 的概念和方法未曾引起人们的注意. 20 世纪 70 年代末, 吴文俊在创立他的几何定理机器证明方法时注意到了 Ritt 的工作, 并以此作为完善其机械化方法的构造性代数工具。吴在理论、算法、效率和实用上都大大发展了特征列方法, 并将其用于各种几何推理和计算问题, 从而引发了大量后续工作, 大量学术论著相继出现。吴方法在于计算多项式组 (而非理想) 的特征列, 它避免了 Ritt 算法中的不可约性限制, 因而使得从任意多项式组有效地构造特征列成为可能。吴—Ritt 特征列方法有着非常广泛的用途, 可以处理很多复杂的问题, 它使得大量繁琐的符号计算问题的机械化实现成为可能。本文旨在利用吴—Ritt 特征列方法来解决求解微分方程中遇到的问题。

考虑如下 KdV 方程

$$u_t + uu_x + \alpha u_{xxx} = 0. \quad (3.1.1)$$

我们先给出行波变换

$$u = u(\xi), \xi = k(x - ct) + \xi_0, \quad (3.1.2)$$

在上述行波变换下, 方程 (3.1.1) 化为:

$$-cu' + uu' + \alpha k^2 u''' = 0. \quad (3.1.3)$$

方程具有如下双曲正切函数多项式的解, 即

$$u(\xi) = \sum_{i=0}^m a_i T^i, T = \tanh(\xi). \quad (3.1.4)$$

将 (3.1.4) 代入 (3.1.3), 使其最高阶导数项 u''' 和非线性项 uu' 的幂次相平衡, 即 $m+3=2m+1$, 由此确定出 $m=2$. 于是可设方程 (3.1.3) 的解为

$$u = a_0 + a_1 T + a_2 T^2 \quad (3.1.5)$$

将 (3.1.5) 式代入 (3.1.3), 合并 T 的同次幂并令各次幂的系数为 0. 消去非零因子后, 我们得到确定 a_0, a_1 和 a_2 的代数方程组

$$\begin{cases} P_1 = (2k^2\alpha + c - a_0)a_1 = 0, \\ P_2 = a_1^2 + 16k^2\alpha a_2 + 2ca_2 - 2a_0a_2 = 0, \\ P_3 = a_1(8k^2\alpha + c - a_0 + 3a_2) = 0, \\ P_4 = -a_1^2 + 40k^2\alpha a_2 + 2ca_2 - 2a_0a_2 + 2a_2^2 = 0, \\ P_5 = a_1(2k^2\alpha + a_2) = 0, \\ P_6 = 12k^2\alpha + a_2 = 0. \end{cases}$$

(将微分方程转化为代数方程组的机器实现已有人作出过相关研究,读者可以在文献[4, 66]中找到有关论述。)

此代数方程组的求解,成为我们求解微分方程过程中的一大难题。为了便于计算,我们利用吴—Ritt 特征列方法。为此,我们先给出吴—Ritt 特征列的算法。吴—Ritt 特征列的算法如下:

算法 BasSet: $BS := BasSet(PS)$. 任给非空多项式组 $PS \subset K[x]$, 本算法计算 PS 的基列 BS .

B1. 命 $FS := PS, BS := \emptyset$.

B2. 重复下列步骤直至 $FS := \emptyset$.

B2. 1. 从 FS 中选取一秩最低的多项式 B .

B2. 2. 命 $BS := BS \cup \{B\}$

B2. 3. 如果 $cls(B) = 0$, 则命 $FS := \emptyset$; 否则命

$$FS := \{F \in FS \setminus \{B\} \mid F \text{ 对 } B \text{ 是约化的}\}.$$

下面我们描述吴—Ritt 的算法, 该算法指出了如何从任意多项式求得其特征列。

算法 CharSet: $CS := CharSet(PS)$. 任给非空多项式组 $PS \subset K[x]$, 本算法计算 PS 的特征列 CS .

C1. 命 $FS := PS, RS := PS$.

C2. 重复下列步骤直至 $RS := \emptyset$.

C2. 1. 计算 $CS := BasSet(FS)$.

C2. 2 如果 CS 是矛盾列, 则命 $RS := \emptyset$; 否则计算

$$RS := prem(FS \setminus CS, CS) \setminus \{0\},$$

且命 $FS := FS \cup RS$.

用吴—特征列消元方法计算特征列的机器实现见附录 2。

令 $PS = [P_1, P_2, \dots, P_6]$. 用吴—特征列消元方法可求得 PS 的特征列为

$$CS = [8\alpha k^2 + c - a_0, a_1, 12\alpha k^2 + a_2].$$

由于 CS 中的多项式的初式均为非零常数, 故 PS 的零点集与 CS 的零点集相同. 容易求得 CS 关于 a_0, a_1, a_2 的零点如下:

$$a_0 = c + 8\alpha k^2, a_1 = 0, a_2 = -12\alpha k^2.$$

于是方程的 (3.1.1) 孤立波解为:

$$u(x, t) = c + 8\alpha k^2 - 12\alpha k^2 \tanh^2[k(x - ct)].$$

以上求解过程借助了计算机代数系统 Maple 环境下的“数学机械化自动推理平台”来解决复杂的代数方程组的求解难题, 这依靠手工过程是很难实现的, 而且求解过程非常的简便快速. 与手工求解过程相比具有极大的优越性.

3.2 一种新的符号计算方法

文献[53, 54, 55]给出了 Adomian 分解方法的原理. 我们在此对之加以引用, 通过添加算法和程序, 给出了一类奇异初值问题的机器实现方法, 解决了用手工过程无法实现的问题.

3.2.1 问题的提出

Adomian 分解方法有着广泛的应用. 此方法给出了快的收敛级数解. 考虑如下奇异初值问题:

$$\begin{aligned} y'' + \frac{n}{x} y' + f(y) &= 0, 0 < x \leq 1, \\ y(0) = A, y'(0) &= B. \end{aligned} \quad (3.2.1)$$

另一类奇异初值问题形如:

$$\begin{aligned} y'' + \frac{n}{x} y' + f(x, y) &= g(x), 0 < x \leq 1, \\ y(0) = A, y'(0) &= B. \end{aligned} \quad (3.2.2)$$

其中 A, B 是常数, $f(x, y)$ 是连续实值函数, 且 $g(x) \in C[0, 1]$.

我们不妨将 (3.2.2) 化归为如下的奇异初值问题:

$$\begin{aligned} [h(x)y'(x)]' + f(x, y(x)) &= g(x), \\ y(0) = A, y'(0) &= B. \end{aligned} \quad (3.2.3)$$

其中 A, B 是常数, $f(x, y)$ 是连续实值函数, $h(x)$ 是连续函数, 且 $h(x) \neq 0$,

$g(x) \in C[0, 1]$. 如果 $h(x)$ 是可微函数, 这时 (3.2.3) 变为:

$$y^n(x) + \frac{h'(x)y'(x)}{h(x)} + F(x, y(x)) = G(x), \quad (3.2.4)$$

$$y(0) = A, y'(0) = B.$$

$$\text{其中 } F(x, y(x)) = \frac{f(x, y(x))}{h(x)}, G(x) = \frac{g(x)}{h(x)}.$$

我们考虑形如(3.2.4)这类奇异初值问题的求解问题, 在求解过程中我们必须考虑如何消掉奇异性的影响, 而又便于我们求解。为此, 我们引入 Adomian 分解方法来求解此类奇异初值问题。

3.2.2 Adomian 分解方法原理

下面我们利用 Adomian 分解方法来计算此类微分方程。为了克服奇异性的影响, 我们定义微分算子 L , 将(3.2.3)重新写成:

$$Ly = -f(x, y) + g(x)$$

微分算子定义为:

$$L = \frac{d}{dx} \left(h(x) \frac{d}{dx} \right).$$

其逆算子被定义为:

$$L^{-1}(\cdot) = \int_0^x h^{-1}(x) \left(\int_0^x (\cdot) dx \right) dx. \quad (3.2.5)$$

将之作用于(3.2.3), 得到:

$$y(x) = A + Bx + L^{-1}(g(x)) - L^{-1}f(x, y). \quad (3.2.6)$$

在 Adomian 分解方法中, $y(x)$ 表示为:

$$y(x) = \sum_{n=0}^{\infty} y_n(x) \quad (3.2.7)$$

非线性函数 $f(x, y)$ 表示为:

$$f(x, y) = \sum_{n=0}^{\infty} A_n \quad (3.2.8)$$

其中 A_n 为 Adomian 多项式, 它与非线性函数 $F(u)$ 有关, 其前几项如下:

$$\begin{aligned}
 A_0 &= F(u_0), \\
 A_1 &= u_1 F'(u_0), \\
 A_2 &= u_2 F'(u_0) + \frac{u_1^2}{2!} F''(u_0), \\
 A_3 &= u_3 F'(u_0) + u_1 u_2 F''(u_0) + \frac{u_1^3}{3!} F'''(u_0), \\
 &\vdots
 \end{aligned}
 \tag{3.2.9}$$

将 (3.2.7) 和 (3.2.8) 代入到 (3.2.6) 中, 我们得到:

$$\sum_{n=0}^{\infty} y_n(x) = A + Bx + L^{-1}(g(x)) - L^{-1}\left(\sum_{n=0}^{\infty} A_n\right).$$

为了决定 $y_n(x)$, 我们使用 Adomian 分解方法, 我们得到如下递归关系式:

$$\begin{aligned}
 y_0(x) &= A + Bx + L^{-1}(g(x)), \\
 y_{k+1}(x) &= -L^{-1}(A_k), k \geq 0.
 \end{aligned}$$

此即为:

$$\begin{aligned}
 y_0(x) &= A + Bx + L^{-1}(g(x)), \\
 y_1(x) &= -L^{-1}(A_0), \\
 y_2(x) &= -L^{-1}(A_1), \\
 &\vdots
 \end{aligned}
 \tag{3.2.10}$$

这样将 (3.2.9) 与 (3.2.10) 联立, 我们便递归地得到 $y_n(x)$, 从而求得 $y(x)$.

对于奇异初值问题 (3.2.4), 可微算子和其逆算子分别为:

$$\begin{aligned}
 L &= h^{-1}(x) \frac{d}{dx} (h(x) \frac{d}{dx}) \\
 L^{-1}(\cdot) &= \int_0^x h^{-1}(x) \left(\int_0^x h(x)(\cdot) dx \right) dx.
 \end{aligned}$$

从而方程 (3.2.4) 的解可以由相应的方法得到。

3.2.3 相关算法

这里我们给出利用 Adomian 分解方法求解奇异初值问题 (3.2.3) 和 (3.2.4) 的相关算法.

算法:

输入: $(h(x), f(x, y), g(x), a, b)$;

输出: 给定微分方程的级数解.

(i) 计算 $L^{-1}(g(x))$,

(ii) 寻找 $y_0 = a + bx + L^{-1}(g(x))$,

(iii) 计算 (3.2.9) 中所描述的 A_0 ,

(iv) 计算出 $y_{i+1}(x) = -L^{-1}(A_i), i = 0, 1, 2, \dots$, 其中 A_{i+1} 可以由 (3.2.9) 递归的计算得到,

(v) 求得解 $y(x) = \sum_{i=0}^n y_i$ (其中 n 可以自定)。

Adomian 分解方法的程序见附录 1。

定理 3.2.3.1 如果由双积分给出的逆算子 $L^{-1}(g(x))$ 存在, 那么算法 (3.2.3) 可以计算由 (3.2.3) 和 (3.2.4) 组成的奇异初值问题的解。

证明: 假定由双积分给出的逆算子 $L^{-1}(g(x))$ 存在, 那么算法 (3.2.3) 中的 (i)

可以计算出 $L^{-1}(g(x))$ 。而算法 (3.2.3) 中的其余部分都基于 Adomian 分解原理。

联立 (3.2.10) 和 (3.2.9), 我们可以递归地决定 $y_n(x)$, 于是我们求得 (3.2.7)

中的 $y(x)$ 。

3.2.4 下面我们给出几个例子来演示算法 (3.2.3) 的应用。

例 1.5.1 考虑如下线性奇异初值问题:

$$\begin{aligned} y'' + 2y' + y &= 0, \\ y(0) &= 1, y'(0) = 0. \end{aligned}$$

计算机输出此初值问题的解为:

$$y(x) = 1 - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{8}x^4 + \frac{1}{30}x^5 + O(x^6).$$

而此方程的闭形式解为:

$$y(x) = e^{-x} + xe^{-x}.$$

例 1.5.2 考虑如下非线性奇异初值问题:

$$\begin{aligned} y'' + y' - 2(x+1)y &= 4y \ln y, \\ y(0) &= 1, y'(0) = 0. \end{aligned}$$

计算机输出的此初值问题的解为:

$$y(x) = 1 + x^2 + \frac{1}{2}x^4 + \frac{1}{6}x^6 + \frac{1}{24}x^8 + O(x^{10}).$$

而此方程的闭形式解为:

$$y(x) = e^{x^2}.$$

在求解过程中, 为了消除奇异性的影响, 我们采用了 Adomian 分解方法。在 Maple 环境下, 我们编制相应的算法和程序来完成整个求解过程, 实现了求解过

程的机械化, 解决了用手工过程无法实现的求解难题。

3.3 用对称变换的方法求解微分方程

因为用对称变换方法求解(偏)微分方程已经有很多人作出过研究, 且取得了不少成果, 相关研究方向已成为一大领域, 限于篇幅, 我们在这里只简要给出其在计算机代数系统 Maple 环境下的机器演示过程和方法, 不作详细知识介绍, 有兴趣的读者可以去研读文献[26, 44, 45, 46, 47, 48]中的有关内容。

输入命令:

```
symgen(ODE, y(x), way = xxx, HINT = ...)
```

参数含义:

<i>ODE</i>	普通微分方程
<i>y(x)</i>	(可选择的) 独立变量
<i>way = xxx</i>	可选择的算法, 决定对称有限生成元的相关系数
<i>HINT = [e₁, e₂]</i>	(可选择的) e ₁ , e ₂ 表示有限生成元的可能的函数形式
<i>HINT = [[e₁, e₂], [e₃, e₄], ...]</i>	(可选择的) 有限生成元的可能函数形式的列表。

```
> with(DEtools):
```

```
> odepde(diff(y(x), x)=phi(x, y(x)));
```

```
(diff(_eta(x, y), x))+((diff(_eta(x, y), y))-(diff(_xi(x, y), x)))
```

```
*phi(x, y)-(diff(_xi(x, y), y))*phi(x, y)^2-_xi(x, y)
```

```
*(diff(phi(x, y), x))-_eta(x, y)*(diff(phi(x, y), y))
```

一阶常微分方程

Kamke's ODE 120

```
> PDEtools[declare](y(x), prime=x);
```

```
> infolevel[symgen] := 2; # turn ON userinfo on symmetries
```

```
infolevel[symgen] := 2
```

```
> ode1 := x*(diff(y(x), x))-y(x)*(x*ln(x^2/y(x))+2) = 0
```

```
ode1 := x*(diff(y(x), x))-y(x)*(x*ln(x^2/y(x))+2) = 0
```

```
> symgen(ode1);
```

```
[_xi = 1, _eta = 2*y/x]
```

Kamke's ODE 236

```
> ode2 := x*(y(x)+4)*diff(y(x), x)-y(x)^2-2*y(x)-2*x = 0;
```

```
ode2 := x*(y(x)+4)*(diff(y(x), x))-y(x)^2-2*y(x)-2*x = 0
```

```
> odeadvisor(ode2);
```

```

      [_rational, [_Abel, `2nd type`, `class B`]]
> sym2 := symgen(ode2);
      sym2 := [_xi = 0, _eta = (-y+x)*(x-2*y-4)/(y+4)],
      [_xi = 0, _eta = -(4*x+y^2)*(-y+x)/(x*(y+4))]
Kamke's ODE 357 (no classification)
> ode3 := x*diff(y(x), x)*ln(x)*sin(y(x))+cos(y(x))*(1-x*cos(y(x))) = 0;
ode3 := x*(diff(y(x), x))*ln(x)*sin(y(x))+cos(y(x))*(1-x*cos(y(x))) = 0
> sym3:=symgen(ode3, way=5);
      sym3 := [_xi = 0, _eta = (cos(2*y)+1)/(ln(x)*sin(y))], [_xi = 0, _eta =
      (-2*ln(x)*cos(y)+x*cos(2*y)*x)/(ln(x)*sin(y))]
> dsolve(ode3);
      y(x) = arccos(ln(x)/(x+_C1))

```

Here is an ODE with an arbitrary function $F(x, y)$ (no classification):

```

> ode4 := diff(y(x), x) = F(y(x)^(3/2)-3/2*exp(x))/y(x)^(1/2)*exp(x);
ode4 := diff(y(x), x) = F(y(x)^(3/2)-3/2*exp(x))*exp(x)/y(x)^(1/2)
> symgen(ode4, way=abaco2);
      [_xi = exp(-x), _eta = 1/y^(1/2)]

```

二阶常微分方程

Kamke's nonlinear second order example number 33;

```

>k33:=diff(y(x), x, x)+(y(x)+3*f(x))*diff(y(x), x)y(x)^3+y(x)^2*f(x)
+ y(x)*(diff(f(x), x) + 2*f(x)^2)=0;
k33 := (diff(y(x), `$(x, 2)))+(y(x)+3*f(x))*(diff(y(x),
x))-y(x)^3+y(x)^2*f(x)+y(x)*((diff(f(x), x))+2*f(x)^2) = 0
> sym33 := symgen(k33, y(x), way=formal);
-> Computing symmetries using: way = formal
[1/exp(Int(-f(x), x)), -f(x)*y/exp(Int(-f(x), x))], [Int(exp(Int(-f(x),
x)), x)/exp(Int(-f(x), x)), -(exp(Int(-f(x), x))+f(x)*Int(exp(Int(-f(x),
x)), x))*y/exp(Int(-f(x), x))]
<- successful computation of symmetries.
sym33 := [_xi = exp(-Int(-f(x), x)), _eta = -f(x)*y*exp(-Int(-f(x), x))],
[_xi = Int(exp(Int(-f(x), x)), x)*exp(-Int(-f(x), x)), _eta =
-y*(1+exp(-Int(-f(x), x))*f(x)*Int(exp(Int(-f(x), x)), x))]

```

This example shows an ODE containing an arbitrary function G :

```

> ode5 := diff(y(x), x, x) =
-4*diff(y(x), x)^2/(4*y(x)-G(diff(y(x), x)*x-1/2*y(x)));
      ode5 := diff(y(x), `$(x, 2)) = -4*(diff(y(x),
      x))^2/(4*y(x)-G(x*(diff(y(x), x))-1/2*y(x)))
> odeadvisor(ode5);
      [[_2nd_order, _with_linear_symmetries]]
This one has two pairs of infinitesimals for ode5 (a point and a dynamical
symmetry):
> sym5 := symgen(ode5);      # _y1 = dy/dx
`` , `-> Computing symmetries using: way = 3`

      [x, 0]
`` , `<- successful computation of symmetries.`
`` , `-> Computing symmetries using: way = 2`

`` , `-> Computing symmetries using: way = abacol`

      [x, 0], [0, 1/_y1]
`` , `<- successful computation of symmetries.`

      sym5 := [_xi = x, _eta = 0], [_xi = 0, _eta = 1/_y1]
> ode6 := diff(y(x), x, x) =
1/x^2*(diff(y(x), x)^2*x^2-2*y(x)*diff(y(x), x)*x+y(x)^2);
ode6 := diff(y(x), `$(x, 2)) = ((diff(y(x), x))^2*x^2-2*y(x)*(diff(y(x),
      x)*x+y(x)^2)/x^2
> symgen(ode6);
-> Computing symmetries using: way = 3
      [0, x], [x, 0]
<- successful computation of symmetries.
      [_xi = 0, _eta = x], [_xi = x, _eta = 0]
Below, G is again an arbitrary function of its arguments (the ODE does
not match any recognizable pattern).
> ode7 := diff(y(x), x, x) =
1/ln(y(x))*(diff(y(x), x)^2 - y(x)*ln(y(x))*diff(y(x), x)

```

```

+ y(x)*ln(y(x))^2*exp(-2*x)*G(diff(y(x), x)/ln(y(x))*exp(x))/y(x);
ode7 := diff(y(x), `$(x, 2)) = ((diff(y(x),
x))^2-y(x)*ln(y(x))*(diff(y(x),
x))+y(x)*ln(y(x))^2*exp(-2*x)*G((diff(y(x),
x))*exp(x)/ln(y(x)))))/(ln(y(x))*y(x))
> odeadvisor(ode7);
[NONE]
> symgen(ode7, way=abacol);
-> Computing symmetries using: way = abacol
[0, ln(y)], [exp(x), 0]
<- successful computation of symmetries.
[_xi = 0, _eta = ln(y)], [_xi = exp(x), _eta = 0]

```

• 高阶常微分方程

A third order example

```

> ode8 := diff(y(x), x, x, x) = 13/8/x^3*y(x)-1/8*(-
20*diff(y(x), x, x)*x^2+26*diff(y(x), x)*x)/x^3-
1/8*(24*diff(y(x), x)*x^3*diff(y(x), x, x)
-20*diff(y(x), x)^2*x^2)/x^3/y(x);
ode8 := diff(y(x), `$(x, 3)) = 13/8*y(x)/x^3-1/8*(-20*(diff(y(x), `$(x,
2)))*x^2+26*x*(diff(y(x), x)))/x^3-1/8*(24*(diff(y(x),
x))*x^3*(diff(y(x), `$(x, 2)))-20*(diff(y(x), x))^2*x^2)/(x^3*y(x))
> odeadvisor(ode8);
[[_3rd_order, _with_linear_symmetries]]
> sym8 := symgen(ode8);
-> Computing symmetries using: way = 2
[0, 1/8*x/y]
<- successful computation of symmetries.
-> Computing symmetries using: way = 3
[0, 1/8*x/y], [0, y], [x, 0]
<- successful computation of symmetries.
sym8 := [_xi = 0, _eta = y], [_xi = x, _eta = 0], [_xi = 0, _eta = 1/8*x/y]
Here is how to find a solution for ode8 using these symmetries:
> ans8 := dsolve(ode8);
ans8:=y(x)=2/29*(145*29^(1/2)*x^(9/4+1/4*29^(1/2))*_C3-145*29^(1/2)*x

```

$$\begin{aligned} & \sqrt[9/4-1/4*29^{(1/2)}]{_C2+1682*_C1*x-841*x} \sqrt[9/4-1/4*29^{(1/2)}]{_C2-841*x} \\ & \sqrt[9/4+1/4*29^{(1/2)}]{*_C3}^{(1/2)}, \\ y(x) = & -2/29*(145*29^{(1/2)}*x^{(9/4+1/4*29^{(1/2)})*_C3-145*29^{(1/2)}*x^{(9/4} \\ & -1/4*29^{(1/2)}*_C2+1682*_C1*x-841*x^{(9/4-1/4*29^{(1/2)})*_C2-841*x^{(9/4} \\ & +1/4*29^{(1/2)}*_C3)^{(1/2)} \end{aligned}$$

用对称变换方法求解微分方程,无论是手工操作还是计算机操作,都有相关的研究成果和文献。但用手工操作几乎只能求其比较简单的常微分方程的解,对于大多数偏微分方程或者比较复杂的高阶常微分方程的求解,其过程相当繁琐,少有人问津。借助计算机代数系统中的Maple软件,轻松便捷的求得很多微分方程的解。这也是用计算机求解的一大优势。

符号计算方法求解微分方程的比较通用的方法还有Groebner基方法,此方法出现的比较早,相关研究成果和文献也很多,限于篇幅,此文中不再介绍有关内容。

第四章 数值方法

文献[29, 33, 34, 35, 36, 37, 38]给出了微分一代数方程(组)的幂级数解和 Pade 近似解的研究工作,我们在研究文献中的原有研究成果的基础上,通过添加新的程序来计算微分一代数方程的幂级数解,完成了求解过程的机器实现,这比原有利用手工过程求得的结果具有更高的精度。

在本章,我们研究了利用计算机代数系统 Matlab 软件通过编制算法和程序来实现微分代数方程的机器求解,并将其解扩展到任意次的幂级数展开法,接着我们再利用计算机代数系统 Maple 软件把幂级数解转化成 Pade 级数近似解形式,最后我们利用 Maple 软件作出 Pade 近似解和精确解在同一坐标系下的图形,给出了微分方程的 Pade 近似解和精确解的逼近比较。

4.1 知识介绍

一类微分代数方程有如下形式:

$$\begin{aligned} F(y, y', x) &= 0 \\ y(x_0) &= y_0, y'(x_0) = y_1. \end{aligned} \quad (4.1.1)$$

其中 F, y 我们假定为足够可微的向量函数,初始值为常数。这样便有:

$$F(y_0, y_0', x_0) = 0. \quad (4.1.2)$$

我们假定 (4.1.1) 的解为:

$$y = y_0 + y_1 x + e x^2 \quad (4.1.3)$$

其中 e 是一个与 y_0 和 y_0' 维数相匹配的向量,我们将 (4.1.3) 代入到 (4.1.2) 中,并且忽略高次项,我们得到如下关于 e 的线性方程:

$$Ae = B \quad (4.1.4)$$

其中 A, B 均为常数矩阵。求解 (4.1.4), 那么方程 (4.1.3) 中 x^2 的相关系数被确定,在高次项下重复上面的过程,我们得到方程 (4.1.1) 的任意次数的幂级数解。而由上述过程给出的幂级数解可以转化为 Pade 级数解,于是我们便得到微分代数方程 (4.1.1) 的任意次数的数值解。

4.2 用幂级数展开法解微分代数方程

我们以如下形式定义另一类型的幂级数解:

$$f(x) = f_0 + f_1x + f_2x^2 + \cdots + (f_n + p_1e_1 + \cdots + p_me_m)x^n \quad (4.2.1)$$

其中 p_1, p_2, \dots, p_m 为常数, e_1, e_2, \dots, e_m 为向量 e 的基, m 是向量 e 的维数. y 是 (4.1.3) 中 m 维的向量, 每一维向量都能被 (4.2.1) 中的幂级数解重复. 因此我们有:

$$y_i = y_{i,0} + y_{i,1}x + y_{i,2}x^2 + \cdots + e_ix^n \quad (4.2.2)$$

由 (4.1.3) 出发, 其中 y_i 是 y 的第 i 维分量, 我们将 (4.2.2) 代入到 (4.1.1) 中, 得到:

$$f_i = (f_{i,n} + p_{i,1}e_1 + \cdots + p_{i,m}e_m)x^{n-j} + Q(x^{n-j+1}) \quad (4.2.3)$$

其中 f_i 为 (4.1.1) 中解 $f(y, y', x)$ 的第 i 维分量, 如果 $f(y, y', x)$ 有 y' , 则 $j=0$, 否则 $j=1$.

由 (4.2.3) 和 (4.1.4), 我们得到 (4.1.4) 中的线性方程如下:

$$\begin{aligned} A_{i,j} &= P_{i,j} \\ B_i &= -f_m \end{aligned} \quad (4.2.4)$$

求解此线性方程, 我们求得 $e_i (i=1, 2, \dots, m)$. 将 e_i 代入到 (4.2.2), 我们得到 $y_i (i=1, 2, \dots, m)$, 其中 y_i 为次数为 n 的多项式, 重复从 (4.2.2) 到 (4.2.4) 的过程, 我们得到微分方程 (4.1.1) 的任意次的幂级数解.

若取 x 处的步长为 h , 将其代入到 y 和 y 的导数中, 我们得到在 $x = x_0 + h$ 处的 y 和 y' 的幂级数解. 如果我们重复上面的过程, 我们得到微分代数方程 (4.1.1) 的数值解, 但我们并不想获得微分方程的幂级数解, 我们可以将它转化为 Pade 级数解的形式.

4.3 Pade 近似解

$$\text{假定 } f(x) = \sum_{i=0}^{\infty} c_i x^i. \quad (4.3.1)$$

同时我们预先假定 $c_j = 0, 1, 2, \dots$, $f(x)$ 为复合函数。Pade 近似解为如下的分数形式:

$$[L/M] = \frac{a_0 + a_1x + \dots + a_Lx^L}{b_0 + b_1x + \dots + b_Mx^M} \quad (4.3.2)$$

它有一个与 (4.3.1) 尽可能一致的麦克劳林展开式, 注意到在 (4.3.2) 中有 $L+1$ 个分子系数, 在 (4.3.2) 中有 $M+1$ 个分母系数, 在它们中或多或少的存在不相关的共同因子, 我们取 $b_0 = 1$. 这种选择是精确决定中非常重要的一部分。于是便有 $L+1$ 个独立的分子系数和 M 个独立的分母系数, 总共 $L+M+1$ 个未知系数, 这些数字表明通常情况下 $[L/M]$ 应该与幂级数 (4.3.1) 通过常规的幂级数解中的一系列的 $1, x, x^2, \dots, x^{L+M}$ 对应, 且有:

$$\sum_{i=0}^{\infty} c_i x^i = \frac{a_0 + a_1x + \dots + a_Lx^L}{b_0 + b_1x + \dots + b_Mx^M} + Q(x^{L+M+1}). \quad (4.3.3)$$

变形得:

$$(b_0 + b_1x + \dots + b_Mx^M)(c_0 + c_1x + \dots) = a_0 + a_1x + \dots + a_Lx^L + Q(x^{L+M+1}). \quad (4.3.4)$$

比较 $x^{L+1}, x^{L+2}, \dots, x^{L+M}$ 的相关系数, 我们有:

$$\begin{aligned} b_M c_{L-M+1} + b_{M-1} c_{L-M+2} + \dots + b_0 c_{L+1} &= 0, \\ b_M c_{L-M+2} + b_{M-1} c_{L-M+3} + \dots + b_0 c_{L+2} &= 0, \\ \vdots & \\ b_M c_L + b_{M-1} c_{L+1} + \dots + b_0 c_{L+M} &= 0. \end{aligned} \quad (4.3.5)$$

如果 $j < 0$, 限定 $c_j = 0$ 为常数. 既然 $b_0 = 1$, 方程 (4.3.5) 变成如下关于 M 的含有 M 个分母相关系数的线性方程组:

$$\begin{bmatrix} c_{L-M+1} & c_{L-M+2} & c_{L-M+3} & \dots & c_L \\ c_{L-M+2} & c_{L-M+3} & c_{L-M+4} & \dots & c_{L+1} \\ c_{L-M+3} & c_{L-M+4} & c_{L-M+5} & \dots & c_{L+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c_L & c_{L+1} & c_{L+2} & \dots & c_{L+M+1} \end{bmatrix} \begin{bmatrix} b_M \\ b_{M-1} \\ b_{M-2} \\ \vdots \\ b_1 \end{bmatrix} = - \begin{bmatrix} c_{L+1} \\ c_{L+2} \\ c_{L+3} \\ \vdots \\ c_{L+M} \end{bmatrix} \quad (4.3.6)$$

于是 b_j 便可以求得. 而比较 (4.3.4) 中 $1, x, x^2, \dots, x^L$ 的相关系数, 得到下列方程组:

$$\begin{aligned}
 a_0 &= c_0, \\
 a_1 &= c_1 + b_1 c_0, \\
 a_2 &= c_2 + b_1 c_1 + b_2 c_0, \\
 &\vdots \\
 a_L &= c_L + \sum_{i=1}^{\min(L,M)} b_i c_{L-i}.
 \end{aligned} \tag{4.3.7}$$

从中求得 a_i . 于是 (4.3.6) 和 (4.3.7) 决定了 Pade 级数的分子和分母, 它们被称作 Pade 方程, 我们建立了 $[L/M]$ 的 Pade 近似解, 这与通过次数 x^{L+M} 得到的

$$\sum_{i=0}^{\infty} c_i x^i \text{ 一致.}$$

4.4 数值例子

例 1 考虑下面的微分代数方程:

$$\begin{bmatrix} 1 & -x \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1' \\ v_2' \end{bmatrix} + \begin{bmatrix} 1 & -(1+x) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \sin x \end{bmatrix}$$

而且有初值:

$$\begin{bmatrix} v_1(0) \\ v_2(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} v_1'(0) \\ v_2'(0) \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}. \tag{4.4.1}$$

原方程的精确解为:

$$\begin{bmatrix} v_1(x) \\ v_2(x) \end{bmatrix} = \begin{bmatrix} \exp(-x) + x \sin x \\ \sin x \end{bmatrix}.$$

从初始值出发, (4.4.1) 的解可以假定为:

$$\begin{aligned}
 v_1(x) &= y_{0,1} + y_{1,1}x + e_1 x^2 = 1 - x - e_1 x^2, \\
 v_2(x) &= y_{0,2} + y_{1,2}x + e_2 x^2 = x + e_2 x^2.
 \end{aligned} \tag{4.4.2}$$

将 (4.4.2) 代入到 (4.4.1), 并且忽略高次项, 我们有:

$$\begin{aligned}
 (-3 + 2e_1)x + Q(x^2) &= 0, \\
 e_2 x + Q(x^3) &= 0.
 \end{aligned} \tag{4.4.3}$$

这些方程都与 (4.2.3) 有关, 与 (4.2.4) 有关的线性方程以如下形式被给出:

$$Ae = B, \tag{4.4.4}$$

$$\text{其中 } A = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, e = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}.$$

从 (4.4.4) 中我们得到线性方程:

$$\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}.$$

求解此线性方程我们得到:

$$e = \begin{bmatrix} 1.5 \\ 0 \end{bmatrix}$$

$$\begin{aligned} v_1(x) &= 1 - x + 1.5x^2, \\ v_2(x) &= x. \end{aligned} \quad (4.4.5)$$

从 (4.4.5) 出发, 我们可以将 (4.4.1) 的解设为:

$$\begin{aligned} v_1(x) &= 1 - x + 1.5x^2 + e_1x^3, \\ v_2(x) &= x + e_2x^3. \end{aligned} \quad (4.4.6)$$

以如此方式, 我们将 (4.4.6) 代入 (4.4.1), 忽略高次项, 我们得到:

$$\begin{aligned} (0.5 + 3e_1)x^2 + Q(x^3) &= 0, \\ (0.1666667 + e_2)x^3 + Q(x^4) &= 0. \end{aligned} \quad (4.4.7)$$

其中

$$A = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} -0.5 \\ -0.1666667 \end{bmatrix}$$

由 (4.4.7) 我们得到线性方程:

$$\begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} -0.5 \\ -0.1666667 \end{bmatrix}.$$

求解此线性方程, 我们得到:

$$e = \begin{bmatrix} -0.1666667 \\ -0.1666667 \end{bmatrix}$$

因此有:

$$\begin{aligned} v_1(x) &= 1 - x + 1.5x^2 - 0.1666667x^3, \\ v_2(x) &= x - 0.1666667x^3. \end{aligned} \quad (4.4.8)$$

重复上面的过程 (在 Matlab 工具箱中, 我们编写了相应的程序完成将幂级数解展开到任意次的机器实现, 详见附录 3), 我们得到:

$$\begin{aligned} v_1(x) &= 1 - x + 1.5x^2 - 0.1666666667x^3 - 0.1250000000x^4 - 0.8333333333 \times 10^{-2}x^5 \\ &\quad + 0.9722222222 \times 10^{-2}x^6 - 0.1984126984 \times 10^{-3}x^7 - 0.1736111111 \times 10^{-3}x^8 \\ &\quad - 0.2755731922 \times 10^{-5}x^9 + 0.3031305115 \times 10^{-5}x^{10}, \\ v_2(x) &= x - 0.1666666667x^3 + 0.8333333333 \times 10^{-2}x^5 - 0.1984126984 \times 10^{-3}x^7 \\ &\quad + 0.2755731922 \times 10^{-5}x^9. \end{aligned}$$

调用 Maple 软件中的 Pade[] 转化功能, 我们将幂级数解 $v_1(x)$ 可以转化为下面的 Pade 级数解:

$$P=[5/4]=\frac{1-0.6633044x+1.2576713x^2+0.2561786x^3-0.0471293x^4-0.5249978x^5}{1+0.3366955x+0.0943669x^2+0.0121698x^3+0.0046050x^4}$$

幂级数解 $v_2(x)$ 可以转化为下面的 Pade 级数解:

$$q=[5/4]=\frac{x+0.1338383x^3-0.0033128x^5}{1+0.0328282x^2+0.0004509x^4}$$

将步长 h 代入到 v_1, v_2 , 我们得到 $v_1(x), v_2(x)$ 和 $v_1(h), v_2(h)$. 如果幂级数解收敛的非常快, 那么 Pade 级数解中的 v_1, v_2 可以迅速消除. Pade 级数近似解法对大多数微分代数方程而言都是简便有效的, 我们也可以利用此方法, 并借助 Maple 软件来将隐函数展开为幂级数解和 Pade 级数解.

下面我们利用 Maple 软件的绘图功能, 分别作出 $v_1(x)$ 和 $v_2(x)$ 的精确解和 Pade 近似解在同一坐标下的图形, 以观察比较 Pade 近似解与精确解的逼近程度.

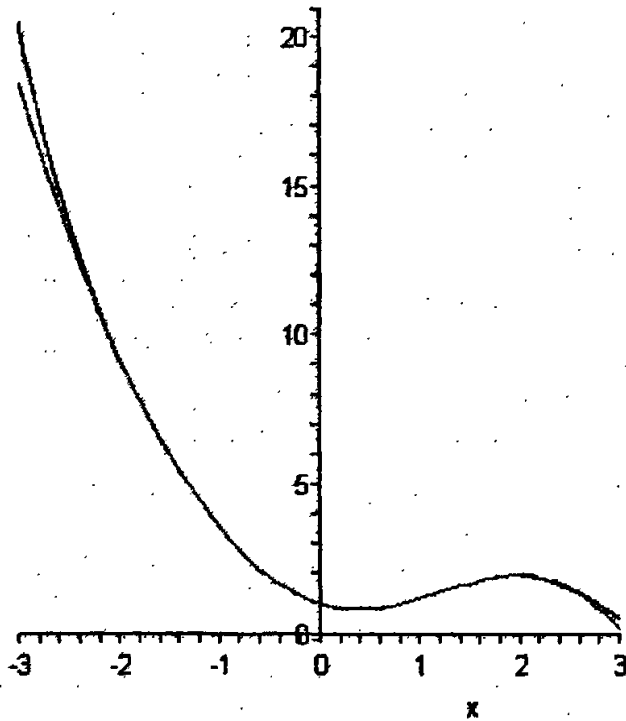


图4-1 $v_1(x)$ 的精确解和其 [5/4] Pade 级数近似解在同一坐标系下的图形

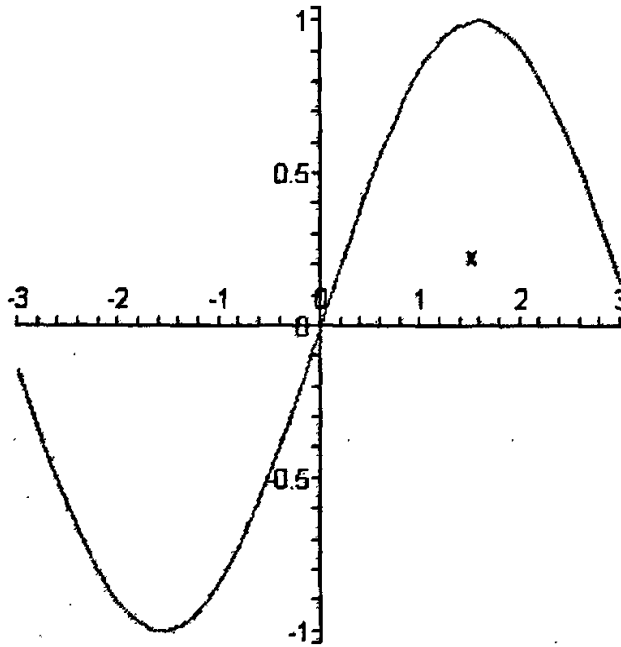


图4-2 $v_2(x)$ 的精确解和其 [5/4] Padé 级数近似解在同一坐标下的图形

用上述同样的方法，我们求得下列微分代数方程：

$$y_1' + y_3 y_2' - (y_2 + 1) y_3' = -y_1 + 1 + \sin x,$$

$$(y_3 + 1) y_1' + y_1 y_2' = -e^{-x},$$

$$y_1 y_2 y_3 - e^{-x} \sin x \cos x = 0.$$

初始值为：

$$y(0) = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, y'(0) = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}.$$

其精确解为：

$$y_1(x) = e^{-x},$$

$$y_2(x) = \sin(x),$$

$$y_3(x) = \cos(x).$$

其幂级数解为：

$$\begin{aligned}
 y_1(x) &= 1 - x + 0.5x^2 - 0.166666667x^3 + 0.0416666667x^4 - 0.00833333333x^5 \\
 &\quad + 0.00138888889x^6 - 0.0001984126984x^7 + 0.00002480158730x^8 - 0.2755731922x^9, \\
 y_2(x) &= x - 0.166666667x^3 + 0.00833333333x^5 - 0.0001984126984x^7 \\
 &\quad + 0.2755731922x^9, \\
 y_3(x) &= 1 - 0.5x^2 + 0.0416666667x^4 - 0.00138888889x^6 + 0.00002480158730x^8.
 \end{aligned}$$

我们将其幂级数解 $y_1(x), y_2(x), y_3(x)$ 转化为如下的 Pade 级数解:

$$\begin{aligned}
 p = [5/4] &= (1 - 0.555555556x + 0.138888889x^2 - 0.01984126984x^3 \\
 &\quad + 0.001653439153x^4 - 0.00006613756614x^5) / (1 - 0.4444444444x \\
 &\quad + 0.08333333333x^2 + 0.007936507937x^3 + 0.0003306878307x^4),
 \end{aligned}$$

$$\begin{aligned}
 q = [5/4] &= (x - 0.1338383838x^3 + 0.003312890813x^5) / (1 + 0.003282828283x^2 \\
 &\quad + 0.0004509379509x^4).
 \end{aligned}$$

$$\begin{aligned}
 r = [5/4] &= (1 - 0.4563492063x^2 + 0.02070105820x^4) / (1 + 0.04365079365x^2 \\
 &\quad + 0.0008597883598x^4).
 \end{aligned}$$

同样, 我们利用 Maple 软件的绘图功能, 分别作出 $y_1(x)$, $y_2(x)$ 和 $y_3(x)$ 的精确解和 Pade 级数近似解在同一坐标下的图形, 来观察比较其 Pade 级数近似解与精确解的逼近程度。

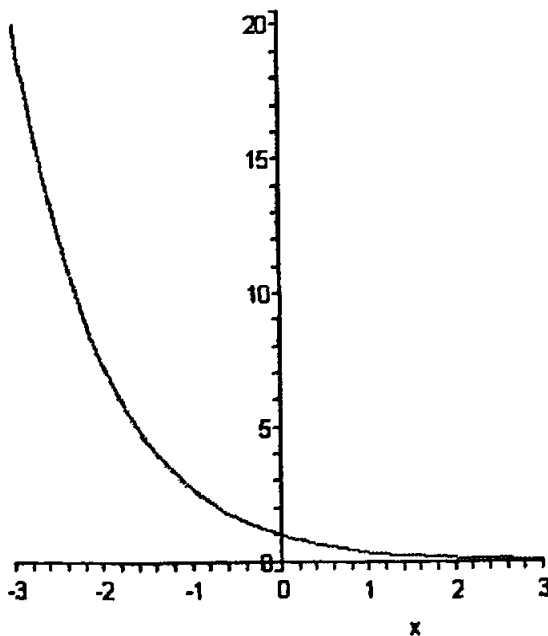


图 4-3 $y_1(x)$ 的精确解和 $[5/4]$ Pade 级数近似解在同一坐标下的图形

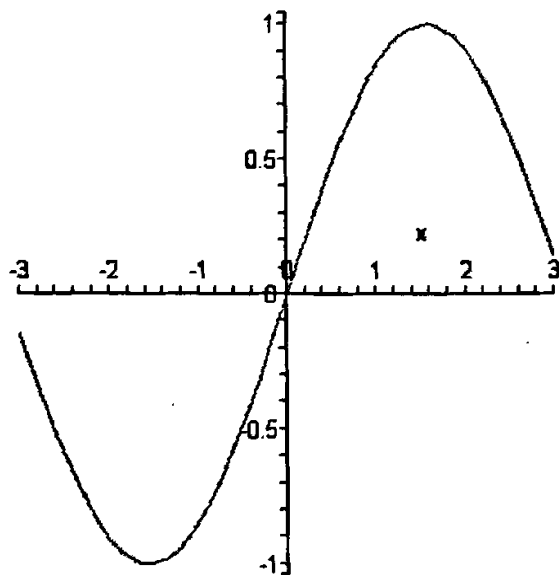


图 4-4 $y_2(x)$ 的精确解和 [5/4]Pade 级数近似解在同一坐标下的图形

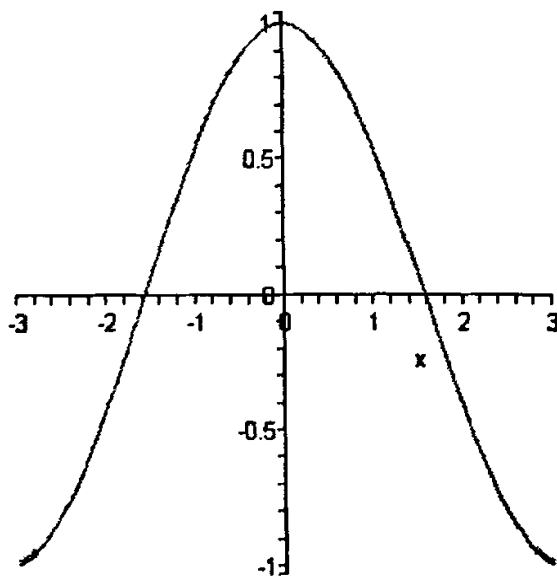


图 4-5 $y_3(x)$ 的精确解和 [5/4]Pade 级数近似解在同一坐标下的图形

由上面给出的精确解和 $[5/4]$ Pade 级数近似解在同一坐标下的图形的比较可以看出,二者逼近程度相当好,由此可见 Pade 级数近似解在微分方程的近似求解中是相当有价值的,值得我们继续研究。

以上求解过程和作图过程都借助了计算机代数系统 (Maple 软件和 Matlab 软件),在 Matlab 符号工具箱中我们编制了求解幂级数的程序,研究了将微分代数的幂级数解展开到任意次的机器实现方法。通过调用 Maple 软件中的 Pade 函数转化功能和作图功能,求得了 Pade 近似数值解,作出了微分代数方程的 Pade 近似数值解和精确解在同一坐标系下的图形。整个操作过程都依靠机器实现。与手工操作过程相比,在计算机上运行操作,简便快速。而与此同时,用手工操作,计算过程非常的繁琐枯燥,可以想象要用手工过程完成上述求解过程,其繁琐是无法想象的,后续计算过程无法开展进行。不仅如此,用计算机求得的结果与手工过程求得的结果相比,计算机求得的结果,精度要高得多,逼近程度好得多。这一点,从上述精确解和 Pade 级数近似解在同一坐标下的图形就可以直观的感受,由此可见,用计算机代数求解微分方程具有极大的优越性,值得研究。

第五章 符号—数值混合计算方法

5.1 方法介绍

在解决实际问题的过程中,有很多复杂的微分方程模型(特别是偏微分方程模型)难于求解。为此,本文在此方面作出了一点探索。首先我们将一个比较复杂的偏微分方程转化为代数方程组(比如利用双曲正切函数展开法),然后对代数方程组中的代数多项式组,利用吴—特征列方法(又称作吴—Ritt 整序原理)加以整理变换,求得此代数多项式组的特征列多项式组,这时原偏微分方程的解就等价于新的特征列多项式方程组的解。如果待求解的特征列方程组形式比较简单(譬如所含变量较少的情形),那么我们就可以直接求得其精确解,如果待求解的特征列方程组为含多变量多参数的复杂代数方程组,那么我们就不能求得其精确解。在实际应用中,在精度允许的范围内,求得难于求解精确解的偏微分方程的近似数值解已经足够。基于此,符号—数值混合计算求解偏微分方程的方法就很值得我们去探索和研究。在已求得原偏微分方程的等价特征列多项式方程组的基础上,我们再引入改进到 Banach 空间的 Newton 迭代法(在 Banach 空间中,对经典 Newton 迭代法加以改进,从而经典 Newton 迭代法中只对连续实值函数成立的情形拓展到对抽象函数也成立),利用 Banach 空间中改进的 Newton 迭代法可以求得等价的待求解的特征多项式方程组的近似数值解,通过对初值和参数的调整,在计算机上运行,就可以求得满足精度要求的近似数值解,(理论上,精度可以依人为设想而具任意精确度,但实际中限于计算机内存只能具备有限精确度,但在一般情况下,这已经符合要求)。从而求得原偏微分方程满足精度要求的近似数值解。

5.2 特征列方法

本节主要讨论特征列方法,本节内容同预备知识一样,主要取之于文献[2]和[3],读者可以从有关文献中找到更详细的论述。

5.2.1 三角列与特征列

设 K 是一特征为零的可计算数域。我们将变元 x 排成固定的次序

$x_1 < x_2 < \dots < x_n$, 对任意多项式 $P \in K[x] \setminus K$, 我们称使得 $\deg(P, x_p) > 0$ 的最大下标 p 为 P 的类, 记为 $cls(P)$. 非零常数的类定义为 0. 设 $p = cls(P) > 0$, 称 x_p 为多项式 P 的导元, 记为 $lv(P)$, $\deg(P, x_p)$ 为 P 的导次数, 记为 $ldeg(P)$, 而 $lc(P, x_p)$ 为 P 的初式, 记为 $ini(P)$.

定义 5.2.1.1 $K[x]$ 中非常数多项式组成的有限非空有序集合

$$T = [T_1, T_2, \dots, T_r]$$

称为三角列或非矛盾拟升列, 如果 $cls(T_1) < cls(T_2) < \dots < cls(T_r)$.

可将任意三角列写成如下形式

$$T = \left[\begin{array}{l} T_1(x_1, \dots, x_n), \\ T_2(x_1, \dots, x_n, \dots, x_{p_2}), \\ \dots \\ T_r(x_1, \dots, x_n, \dots, x_{p_2}, \dots, x_{p_r}) \end{array} \right], \quad (5.2.1)$$

这里

$$\begin{aligned} 0 < p_1 < p_2 < \dots < p_r \leq n, \\ p_i &= cls(T_i), x_{p_i} = lv(T_i), i = 1, \dots, r. \end{aligned}$$

设 T 为 (5.2.1) 所示的三角列, 而 P 为任一多项式. 如果 $\deg(P, x_{p_i}) < ldeg(T_i)$, 对所有的 i 成立, 则称 P 对 TS 是约化的. 将多项式

$$R = prem(\dots prem(P, T_r, x_{p_r}), \dots, T_1, x_{p_1}).$$

简记为 $prem(P, TS)$, 并称其为 P 对 TS 的伪余式. 有如下伪余公式

$$I_1^{q_1} \dots I_r^{q_r} P = \sum_{i=1}^r Q_i T_i + R, \quad (5.2.2)$$

这里 q_i 是非负整数, 而

$$I_i = ini(T_i), Q_i \in K[x], i = 1, \dots, r.$$

明显地, 在 P 对 TS 是约化的情形有 $prem(P, TS) = P$. 对任意多项式组 PS , $prem(PS, TS)$ 代表 $\{prem(P, TS) | P \in PS\}$.

引理 5.2.1.2 对 $K[x]$ 中的任意三角列 TS 与多项式 P , 如果 $prem(P, TS) = 0$, 则 $Zero(TS / ini(TS)) \subset Zero(P)$.

证明参见文献[2], 从略.

定义 5.2.1.3 三角列 $T=[T_1, T_2, \dots, T_r] \subset K[x]$ 称为非矛盾升列, 如果对所有的 $2 \leq i \leq r$, T_i 对 $[T_1, T_2, \dots, T_{i-1}]$ 都是约化的.

每个单个非零常数构成的集合都称为矛盾升列.

定义 5.2.1.4 升列 CS 称为非空多项式组 $PS \subset K[x]$ 的特特征列, 如果

$$CS \subset \langle PS \rangle, \text{prem}(PS, CS) = \{0\}.$$

命题 5.2.1.5 设 $CS=[C_1, \dots, C_r]$ 为多项式方程组 $PS \subset K[x]$ 的特特征列, 且命

$$I_i = \text{ini}(C_i), PS_i = PS \cup \{I_i\}, i=1, \dots, r,$$

$$IS = \text{ini}(CS) = \{I_1, \dots, I_r\}.$$

则

$$\text{Zero}(CS/IS) \subset \text{Zero}(PS) \subset \text{Zero}(CS), \tag{5.2.3}$$

$$\text{Zero}(PS) = \text{Zero}(CS/IS) \cup \bigcup_{i=1}^r \text{Zero}(PS_i) \tag{5.2.4}$$

在 K 以及 K 的任意扩域中成立.

5.2.2 吴—Ritt 算法

定义 5.2.2.1 称 $K[x]$ 中的非零多项式 F 比 G 有较低的秩, 记作 $F < G$ 或 $F > G$, 如果 $\text{cls}(F) < \text{cls}(G)$ 或者 $\text{cls}(F) = \text{cls}(G) > 0$, 且 $l \deg(F) < l \deg(G)$. 这时, 也称 G 比 F 有较高的秩.

如果 $F < G$ 和 $F > G$ 都不成立, 则称 F 和 G 具有相同的秩, 记作 $F \sim G$.

定义 5.2.2.2 对任意三角列

$$T=[T_1, T_2, \dots, T_r], T'=[T'_1, T'_2, \dots, T'_r],$$

如果存在 $j \leq \min(r, r')$ 使得

$$T_1 \sim T'_1, \dots, T_{j-1} \sim T'_{j-1}, \text{ 而 } T_j > T'_j;$$

或者 $r' > r$, 且 $T_1 \sim T'_1, \dots, T_r \sim T'_r$, 则称 T 比 T' 有较高的秩, 记作 $T > T'$ 或 $T' < T$. 此时, 也称 T' 比 T 有较低的秩. 如果 $T < T'$ 和 $T' < T$ 都不成立, 则称 T 和 T' 具有相同的秩, 记作 $T \sim T'$. 这时有

$$r' = r \text{ 且 } T_1 \sim T'_1, \dots, T_r \sim T'_r.$$

考虑任意非空多项式组 PS . 设 Φ 为所有包含于 PS 的升列构成的集合. 由于每个多项式都构成升列, 所以 $\Phi \neq \emptyset$. 称 Φ 的任意极小升列为 PS 的基列, 这样的基列存在, 并按如下步骤进行:

从 $PS = FS_1$ 开始, 我们在其中选取一个秩最低的多项式, 譬如 B_1 . 若 $cls(B_1) = 0$, 则 $[B_1]$ 是 PS 的一基列. 否则, 命

$$FS_2 = \{F \in FS_1 \setminus \{B_1\} \mid F \text{ 对 } B_1 \text{ 是约化的}\}.$$

如果 $FS_2 = \emptyset$, 那么 $[B_1]$ 是 $FS_1 = PS$ 的一基列, 由 B_1 的选取可知, FS_2 中的所有多项式都比 B_1 有较高的秩. 现设 B_2 为 FS_2 中秩最低的多项式, 且命

$$FS_3 = \{F \in FS_2 \setminus \{B_2\} \mid F \text{ 对 } B_2 \text{ 是约化的}\}.$$

若 $FS_3 = \emptyset$, 则 $[B_1, B_2]$ 为 PS 的一基列. 否则, 从 FS_3 中选取一秩最低的多项式 B_3 , 并按上述方式继续进行. 由于

$$cls(B_1) < cls(B_2) < cls(B_3) < \dots \leq n$$

这一程序必在有限步内停止, 而最终获得 PS 的一基列.

下面算法更确切的给出上述选择过程.

算法 BasSet: $BS := BasSet(PS)$. 任给非空多项式组 $PS \subset K[x]$, 本算法计算 PS 的基列 BS .

B1. 命 $FS := PS, BS := \emptyset$.

B2. 重复下列步骤直至 $FS = \emptyset$.

B2. 1. 从 FS 中选取一秩最低的多项式 B .

B2. 2. 命 $BS := BS \cup \{B\}$

B2. 3. 如果 $cls(B) = 0$, 则命 $FS = \emptyset$; 否则命

$$FS := \{F \in FS \setminus \{B\} \mid F \text{ 对 } B \text{ 是约化的}\}.$$

下面我们描述吴—Ritt 特征列的算法, 该算法指出了如何从任意多项式组求得其特征列.

算法 CharSet: $CS := CharSet(PS)$. 任给非空多项式组 $PS \subset K[x]$, 本算法计算 PS 的特征列 CS .

C1. 命 $FS := PS, RS := PS$.

C2. 重复下列步骤直至 $RS = \emptyset$.

C2. 1. 计算 $CS := BasSet(FS)$.

C2. 2 如果 CS 是矛盾列, 则命 $RS = \emptyset$; 否则计算

$$RS := \text{prem}(FS \setminus CS, CS) \setminus \{0\},$$

且命 $FS := FS \cup RS$.

引理 5.2.2.3 设 $PS \subset K[x]$ 是以

$$BS = [B_1, B_2, \dots, B_r]$$

为基列的非空多项式组, 这里 $\text{cls}(B_i) > 0$. 如果 B 是一对 BS 约化的非零多项式,

那么 $PS \cup \{B\}$ 有一基列, 其秩比 BS 的秩低.

证明参见文献[2], 从略.

可将算法 CharSet 图示如下:

$$\begin{array}{ccccccc} PS = & FS_1 & \subset & FS_2 & \subset & \dots & \subset & FS_m \\ & \cup & & \cup & & & \cup & \\ & BS_1 & & BS_2 & & \dots & & BS_m = CS \\ & RS_1 & & RS_2 & & \dots & & RS_m = \emptyset \end{array}$$

其中

$$\begin{aligned} RS_i &= \text{prem}(FS_i \setminus BS_i, BS_i) \setminus \{0\}, \\ FS_{i+1} &= FS_i \cup RS_i, \end{aligned}$$

且 BS_i 是 FS_i 的基列.

整序原理: 设 CS 为多项式 PS 的一个特征列, 则

$$\left. \begin{aligned} \text{Zero}(CS / IP) &\subset \text{Zero}(PS) \subset \text{Zero}(CS), \\ \text{Zero}(PS / IP) &= \text{Zero}(CS / IP), \\ \text{Zero}(PS) &= \text{Zero}(CS / IP) + \cup, \text{Zero}(PS + \{I_i\}). \end{aligned} \right\}$$

此式中 I_i 为 CS 中多项式 C_i 的初式, IP 是 CS 的初式积. 此外, CS 的序不会高于 PS 的序.

5.2.3 多项式组的零点分解

定理 5.2.3.1 (零点分解定理 I) 存在一个算法, 使得对于给定的多项式集 PS , 可在有限步内得到有限个升列 CS_j , 使得

$$\text{Zero}(PS) = \cup_j \text{Zero}(CS_j / IP_j), \tag{I}$$

$$\text{Remdr}(PS / CS_j) = \{0\}. \tag{5.2.5}$$

(I)中的每个 IP_j 为相应升列 CS_j 的初式积。此外, (5.2.5) 中的每个 CS_j 的序都不高于 PS 的序。

定理 5.2.3.2 (零点分解定理 II) 存在一个算法, 使得对于给定的多项式集 PS , 可在有限步内计算出有限多个升列 CS_j , 满足

$$Zero(PS) = \bigcup_k Zero(CS_k / ISP_k), \tag{II}$$

$$Remdr(PS / CS_k) = \{0\}. \tag{5.2.6}$$

(II) 中的每个 ISP_k 是相应的升列 CS_k 的初式隔离子积。此外, (5.2.6) 中的每个 CS_k 的序不高于 PS 的序。

定义 5.2.3.3 上述定理 5.2.3.1 和定理 5.2.3.2 中给出的升列 CS_j 组成的集合称作 PS 的一个特征序列。

5.3 Banach 空间中改进的 Newton 迭代法

文献[32]和[71]研究了精确的 Newton 方法求解 Jacobian 矩阵非奇异的代数方程组的实解, 文献[30]和[74]研究了不精确的 Newton 方法求解 Jacobian 矩阵非奇异的代数方程组的实解, 文献[14]和[15]引入右逆矩阵的思想改进原有 Newton 迭代法, 但所求解的代数方程组的 Jacobian 矩阵依然是非奇异的。我们在研究上述文献研究成果的基础上, 通过引入文献[14]和[15]中的右逆矩阵的思想并对原有已知研究成果加以改进, 给出了新的 Banach 空间中改进的 Newton 迭代法, Banach 空间中改进的 Newton 迭代法便于我们求解代数方程组的 Jacobian 矩阵是非奇异或奇异的情形, 扩大了求解代数方程组的近似数值解的适用范围。

我们在 Banach 中对经典 Newton 方法加以改进。首先我们给出范数的定义:

$$\|x\|_{\infty} = \max_{1 \leq i \leq k} |x_i|,$$

$$\|x\|_1 = \sum_{i=1}^k |x_i|,$$

$$\|x\|_p = \sqrt[p]{\sum_{i=1}^k |x_i|^p} \quad (p > 1).$$

我们定义 F 关于 x 的 Jacobian 矩阵为如下的 $h \times k$ 矩阵:

$$J_F(x) \equiv J(x) = \left(\frac{\partial f_i(x)}{\partial x_j} \right), i=1, \dots, h, j=1, \dots, k. \tag{5.3.1}$$

当向量 $F(x)$ 和 x 维数相同时 (如 $h=k$), 我们假定方程组的 Jacobian 矩阵非奇异, 如 $J(x) \neq 0$. 对于有界闭集 S 中的 x , 我们定义 F 在 x 点的逆 Jacobian 矩阵 $J_F^{-1}(x)$. 在此情形下, 精确 Newton 方法 (如文献 [32, 71]) 或不精确的 Newton 方法 (如文献 [30, 74]) 都是求解方程组实解的常规著名方法. 此时迭代可以写作如下形式:

$$x^{(n+1)} = g(x^{(n)}) = x^{(n)} + s^{(n)}. \quad (5.3.2)$$

其中 $x^{(n)}$ 为当前近似解. $s^{(n)}$ 为精确 Newton 方法中的 Newton 步长或满足如下关系的 Newton 步长的近似

$$\|F(x^n) + J(x^n)s^{(n)}\| \leq \eta^{(n)} \|F(x^n)\|, (\eta^{(n)} \in [0, 1]).$$

当 $F(x)$ 和 x 的维数不同时, 常规 Newton 方法不能再使用, 因为这时方程组的 Jacobian 矩阵奇异甚至非方阵. 从而我们无法利用常规牛顿迭代法和其他传统方法来求解此类非线性代数方程. 为此, 我们引用加号广义逆矩阵来对 Banach 空间中常规 Newton 方法加以改进, 改进后的牛顿法我们称之为 Banach 空间中改进的牛顿法 (以后简称改进的牛顿法), Banach 空间中改进的牛顿法对于常规情形依然成立, 前者为后者的推广.

定义 5.3.1 A 的加号广义逆矩阵 A^+ 满足:

$$AA^+A = A, A^+AA^+ = A^+, (A^+A)^* = A^+A, (AA^+)^* = AA^+. \quad (5.3.3)$$

其中 $(\cdot)^*$ 代表矩阵的共轭转置矩阵.

下面的定理给出了加号广义逆矩阵的存在性和唯一性.

引理 5.3.2 任意矩阵 A , 其加号广义逆矩阵存在且唯一.

证明见文献 [72] 或文献 [73].

定义 5.3.3 A 的加号广义逆矩阵称作 A 的右逆矩阵, 如果满足 $AA^\dagger = I$, 右逆矩阵记作 A^\dagger .

本定义取之于文献 [14, 15].

如果一个矩阵 A 有右逆矩阵, 其右逆矩阵存在且唯一.

引理 5.3.4 $F(x)$ 为一方程组, J 是 F 的 Jacobian 矩阵. 若 $J(x)$ 是行满秩的,

则 $J^\dagger(x) = J^*(JJ^*)^{-1}(x)$ 是 $J(x)$ 的右逆.

证明: 取 R^h 空间中的一组基 $\{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_h\}$,

由 $B^*JJ^*B = (J^*B)^*(J^*B) = 0 \Rightarrow J^*B = 0$.

假设存在 $r_1, r_2, \dots, r_h \in R$, 使得

$$r_1 JJ^* \vec{u}_1 + r_2 JJ^* \vec{u}_2 + \dots + r_h JJ^* \vec{u}_h = 0 \quad .$$

由 $JJ^*(r_1 \vec{u}_1 + \dots + r_h \vec{u}_h) = 0 \Rightarrow J^*(r_1 \vec{u}_1 + \dots + r_h \vec{u}_h) = 0$.

又 $\text{rank}(J^*) = \text{rank}(J) = h$, 于是我们有 $r_1 \vec{u}_1 + \dots + r_h \vec{u}_h = 0$.

由 $\vec{u}_1, \vec{u}_2, \dots, \vec{u}_h$ 的线性无关性知

$$r_1 = r_2 = \dots = r_h = 0.$$

从而 $JJ^* \vec{u}_1, JJ^* \vec{u}_2, \dots, JJ^* \vec{u}_h$ 是线性无关的, 则 $\text{rank}(JJ^*) = h$.

因此 $J^\dagger(x) = J^*(JJ^*)^{-1}(x)$ 是矩阵 $J(x)$ 的右逆矩阵, $J^\dagger(x)$ 是 $k \times h$ 的矩阵.

有了上述知识, 我们给出如下改进的 Newton 迭代方法:

$$g(x) = x + s = x - J^\dagger(x)F(x) = x - J^*(x)(J(x)J^*(x))^{-1}F(x). \quad (5.3.4)$$

其中 $g(x)$ 和 x 是维数相同 k 维向量, 步长 $-J^*(x)(J(x)J^*(x))^{-1}F(x)$ 是 $J_s = -F(x)$ 的极小范数解.

5.3.1 改进 Newton 迭代法的主要算法

算法:

对 $\forall F_i \in F$, 定义:

$$g_i(x) = x - J_i^\dagger(x)F_i(x) = x - J_i^*(x)(J_i(x)J_i^*(x))^{-1}F_i(x).$$

$$X := x^{(0)};$$

$$Y := g_i(X);$$

while $\|Y - X\| \geq \varepsilon$, do

$$X := Y;$$

$$Y := g_i(x);$$

return(Y).

当 $\{x^{(0)}, x^{(1)}, \dots, x^{(n)}, \dots\}$ 收敛到 g 的固定点 α 时, α 在 $x^{(0)}$ 的领域内, 且有 $F(\alpha) = 0$.

当 $h = k$ 时, J^\dagger 和 J^{-1} 存在且 $J^\dagger = J^{-1}$. 这时改进的 Newton 方法类似于常规 Newton 方法, 方程组中含有 n 个方程和 n 个未知数.

5.3.2 算法的正确性和渐进收敛性

给定一个初始点 $\mathbf{x}^{(0)} = [x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}]^T$. 我们建立如下迭代序列:

$$\mathbf{x}^{(n+1)} = g(\mathbf{x}^{(n)}) \quad (5.3.5)$$

其中 g 在 (5.3.4) 中给出. 下面我们考虑算法的渐进收敛性.

定理 5.3.2.1 假定 $g(x)$ 满足:

$$\|g(x) - g(y)\| \leq \lambda \|x - y\|. \quad (5.3.6)$$

对每个向量 x, y 满足 $\|x - x^{(0)}\| \leq \rho, \|y - y^{(0)}\| \leq \rho$, λ 为 Lipschitz 常数, 满足

$$0 \leq \lambda < 1. \quad (5.3.7)$$

让初始迭代满足

$$\|g(x^{(0)}) - x^{(0)}\| \leq (1 - \lambda)\rho. \quad (5.3.8)$$

则有:

$$(i) \text{ 对所有迭代 (5.3.5), 满足 } \|x - x^{(0)}\| \leq \rho. \quad (5.3.9)$$

(ii) 迭代收敛到 g 的某一固定点 α .

(iii) α 为 g 在 $\|x - x^{(0)}\| \leq \rho$ 中的唯一固定点.

(iv) α 为 F 在 $\|x - x^{(0)}\| \leq \rho$ 中的解.

证明:

(i) 已知 $x^{(1)} = g(x^{(0)})$, 由 (5.3.6), (5.3.7) 有:

$$\|x^{(1)} - x^{(0)}\| \leq (1 - \lambda)\rho \leq \rho. \quad (5.3.10)$$

从而 $x^{(1)}$ 满足 (5.3.9), 假定对于迭代 $x^{(1)}, x^{(2)}, \dots, x^{(n)}$, (5.3.9) 均成立, 于是由

(5.3.5) 有:

$$\begin{aligned} \|x^{(n+1)} - x^{(n)}\| &= \|g(x^{(n)}) - g(x^{(n-1)})\| \\ &\leq \lambda \|x^{(n)} - x^{(n-1)}\| \\ &\leq \lambda^2 \|x^{(n-1)} - x^{(n-2)}\| \\ &\vdots \\ &\leq \lambda^n \|x^{(1)} - x^{(0)}\| \\ &\leq \lambda^n (1 - \lambda)\rho. \end{aligned} \quad (5.3.11)$$

从而 $\|x^{(n+1)} - x^{(0)}\| = \|(x^{(n+1)} - x^{(n)}) + \dots + (x^{(1)} - x^{(0)})\|$

$$\begin{aligned} &\leq \|x^{(n+1)} - x^{(n)}\| + \dots + \|x^{(1)} - x^{(0)}\| \\ &\leq (\lambda^n + \lambda^{(n-1)} + \dots + 1)(1 - \lambda)\rho \\ &= (1 - \lambda^{n+1})\rho \\ &\leq \rho. \end{aligned}$$

(i) 得证。

(ii) 对任意正整数 m 和 p , 由

$$\begin{aligned} \|x^{(m+p)} - x^{(m)}\| &= \|(x^{(m+p)} - x^{(m+p-1)}) + \dots + (x^{(m+1)} - x^{(m)})\| \\ &\leq \|x^{(m+p)} - x^{(m+p-1)}\| + \dots + \|x^{(m+1)} - x^{(m)}\| \\ &\leq (\lambda^{m+p-1} + \lambda^{m+p-2} + \dots + \lambda^m)(1 - \lambda)\rho \quad (5.3.12) \\ &\leq (1 - \lambda^p)\rho\lambda^m. \end{aligned}$$

$\forall \varepsilon > 0, 0 < \lambda < 1, \exists N(\varepsilon) > 0, \lambda^N < \varepsilon / \rho$, 对 $\forall m > N(\varepsilon)$ 和 $p > 0$ 均有 $\|x^{(m+1)} - x^{(m)}\| < \varepsilon$ 成立。因此 $\{x^{(n)}\}$ 为 Cauchy 序列, 从而 $\{x^{(n)}\}$ 有极限, 记为 α 。在 (5.3.9) 中定义的集合记为 S 。

既然函数 $g_i(x), i=1, \dots, k$ 在集 S 中连续, $\{g(x^{(n)})\}$ 有极限 $g(\alpha)$, 在 (5.3.5) 中的极限也必为 α , 即 $\alpha = g(\alpha), \|x^{(n)} - \alpha\| = \|g(x^{(n-1)}) - g(\alpha)\| \leq \lambda \|x^{(n-1)} - \alpha\|$ 。

于是 $\|x^{(n)} - \alpha\| = \|g(x^{(n-1)}) - g(\alpha)\| \leq \lambda^n \|x^{(0)} - \alpha\| \leq \rho\lambda^n$ 。(ii) 得证。

(iii) 唯一性。假定 β 是集 S 中的另一根, 即 $\|\alpha - \beta\| \neq 0$ 。由 (5.3.6), (5.3.7), 我们有:

$$\|\alpha - \beta\| = \|g(\alpha) - g(\beta)\| \leq \lambda \|\alpha - \beta\| < \|\alpha - \beta\|.$$

矛盾, 从而 $\alpha = \beta$ 。

(iv) 显然 α 是 F 在 $\|x - x^{(0)}\| \leq \rho$ 中的解, 既然 J 是行满秩的, 从而 $J^*(JJ^*)^{-1}$ 是列满秩的。

定理 5.3.2.1 表明如果我们能找到一个满足定理条件的初始迭代点 $x^{(0)}$, 迭代便可以收敛到 g 的一个固定点。换句话说, 在这种情形下, 迭代的收敛性不仅依赖函数 g 本身, 而且依赖初始迭代。

实际上, 在我们能确信函数 g 收敛到某一固定点时, 我们能在初始迭代点的选取上赋予较少的限制条件。

定理 5.3.2.2 若 $g(x)$ 有一固定点 α , 且 $g_i(x)$ 有连续的一阶偏导数, 满足

$$\left| \frac{\partial g_i(x)}{\partial x_j} \right| \leq \frac{\lambda}{k}, \lambda < 1. \quad (5.3.13)$$

对所有的 x 均有 $\|x - \alpha\| \leq \rho$ (5.3.14)

则有:

(i) 对任意满足 (5.3.14) 的 $x^{(0)}$, (5.3.5) 中的所有迭代点 $x^{(n)}$ 均满足 (5.3.14);

(ii) 对任意满足 (5.3.14) 中的 $x^{(0)}$, 迭代收敛到 $x = g(x)$ 的解 α , 且 (5.3.14) 中的 α 是唯一的。

证明: 对满足 (5.3.14) 中的任意两点 x, y , 由泰勒定理有:

$$g_i(x) - g_i(y) = \sum_{j=1}^k \frac{\partial g_i(\xi^{(i)})}{\partial x_j} (x_j - y_j), i = 1, \dots, k.$$

其中 $\xi^{(i)} \in (x, y)$. 因而 $\xi^{(i)}$ 也在 (5.3.14) 中, 利用 (5.3.13) 有:

$$|g_i(x) - g_i(y)| \leq \sum_{j=1}^k \left| \frac{\partial g_i(\xi^{(i)})}{\partial x_j} \right| |x_j - y_j| \leq \|x - y\|_{\infty} \sum_{j=1}^k \left| \frac{\partial g_i(\xi^{(i)})}{\partial x_j} \right| \leq \lambda \|x - y\|_{\infty}.$$

此不等式对所有的 i 均成立, 从而我们有 $\|g(x) - g(y)\| \leq \lambda \|x - y\|_{\infty}$. 于是我们证得

$g(x)$ 在 (5.3.14) 中关于 ∞ -范数是 Lipschitz 连续的. 对 (5.3.14) 中任意 $x^{(0)}$, 有:

$$\|x^{(1)} - \alpha\|_{\infty} = \|g(x^{(0)}) - g(\alpha)\|_{\infty} \leq \lambda \|x^{(0)} - \alpha\|_{\infty} \leq \lambda^n \rho.$$

从而 $x^{(1)}$ 也是 (5.3.14) 中的点, 依次类推, 我们有:

$$\|x^{(n)} - \alpha\|_{\infty} = \|g(x^{(n-1)}) - g(\alpha)\|_{\infty} \leq \lambda \|x^{(n-1)} - \alpha\|_{\infty} \leq \dots \leq \lambda^n \rho. \quad (5.3.15)$$

于是对 (5.3.14) 中所有的 $x^{(n)}$, 在 $\lambda < 1$ 时收敛性成立。

我们有对其他范数也成立的如下推论。

推论 5.3.2.3 定理 5.3.2.2 对求和范数和 p -范数也成立。

证明: 对求和范数定理 5.3.2.2 显然成立. 对 p -范数, 根据 Hölder 不等式和 (5.3.13) 有:

$$\begin{aligned} |g_i(x) - g_i(y)| &\leq \sum_{j=1}^k \left| \frac{\partial g_i(\xi^{(i)})}{\partial x_j} \right| |x_j - y_j| \\ &\leq \sqrt[q]{\sum_{j=1}^k \left| \frac{\partial g_i(\xi^{(i)})}{\partial x_j} \right|^q} \cdot \sqrt[p]{\sum_{j=1}^k |x_j - y_j|^p} \\ &\leq \frac{\lambda}{\sqrt[q]{k}} \|x - y\|_p. \end{aligned}$$

其中 $p \geq 1, q \geq 1, p+q = pq$.

此不等式对每个 i 都满足, 我们有:

$$\|g(x) - g(y)\|_p \leq \sqrt[p]{\sum_{i=1}^k \left(\frac{\lambda}{\sqrt[q]{k}} \|x - y\|_p \right)^p} \leq \lambda \|x - y\|_p.$$

其他细节的证明与前面的定理类似.

推论 5.3.2.4 在定理 5.3.2.2 的条件下, 改进 Newton 迭代方法是二次收敛的.

证明: 设迭代有一近似解 α , 方程 (5.3.13) 对某一 $\rho > 0$ 成立. 如果还有

$$\frac{\partial g_i(\alpha)}{\partial x_j} = 0, (i, j = 1, \dots, n).$$

二阶偏导 $\frac{\partial^2 g_i(x)}{\partial x_j \partial x_j}$ 在解的领域内存在. 由泰勒定理:

$$g_i(x) - g_i(\alpha) = \frac{1}{2} \sum_{j=1}^k \sum_{j'=1}^k \frac{\partial^2 g_i(\xi^{(i)})}{\partial x_j \partial x_{j'}} (x_j - \alpha_j)(x_{j'} - \alpha_{j'})$$

在迭代过程中, 有 $\|x^{(n)} - \alpha\| \leq M \|x^{(n-1)} - \alpha\|^2$.

其中 M 满足:

$$\max_{i,j,j'} \left| \frac{\partial^2 g_i(x)}{\partial x_j \partial x_{j'}} \right| \leq \frac{2M}{k^2}.$$

故改进的 Newton 迭代法是二次收敛的.

5.4 符号-数值混合计算偏微分方程的算例

考察如下形式的偏微分方程:

$$u_t - u_{xx} - \alpha \frac{1}{u} (u_x)^2 = u(1 - u^\rho), \tag{5.4.1}$$

平衡 (5.4.1) 中的线性最高阶导数项 u_{xx} 与非线性项 $u(1 - u^\rho)$ 的幂次, 可得

$$m+2=(\beta+1)m \Rightarrow m=\frac{2}{\beta}.$$

引进变换 $u=v^{2/\beta}$ 将方程 (5.4.1) 改写为

$$vv_t - vv_{xx} - \left(\frac{2\alpha+2}{\beta}-1\right)v_x^2 = \frac{1}{2}\beta v^2(1-v^2). \quad (5.4.2)$$

此时方程 (5.4.2) 的解的阶数为 1. 方程有如下形式的解:

$$v = a_0 + a_1 T, \quad (5.4.3)$$

$$T = \tanh(\xi). \quad (5.4.4)$$

将 (5.4.3) 代入 (5.4.2), 合并 T 的同次幂并令各幂次系数为 0; 于是得到非线性代数方程组如下:

$$\begin{cases} P_1 = \beta^2 a_0^4 - \beta^2 a_0^2 - 2\beta k c a_0 a_1 - 4k^2 a_1^2 + 2\beta k^2 a_1^2 - 4k^2 \alpha a_1^2 = 0, \\ P_2 = 2k^2 a_0 - \beta a_0 + 2\beta a_0^3 - k c a_1 = 0, \\ P_3 = 2\beta k c a_0 - \beta^2 a_1 + 8k^2 a_1 + 8k^2 \alpha a_1 + 6\beta^2 a_0^2 a_1 = 0, \\ P_4 = k c a_1 - 2k^2 a_0 + 2\beta a_0 a_1^2 = 0, \\ P_5 = \beta^2 a_1^2 - 4k^2 - 2\beta k^2 - 4k^2 \alpha = 0. \end{cases}$$

令 $PS = \{P_1, \dots, P_5\}$. 在变元序 $k < a_0 < a_1 < c$ 之下, PS 的特征列为

$CS = [C_1, \dots, C_4]$, 其中:

$$\begin{cases} C_1 = -\beta^2 + 16k^2 + 8\beta k^2 + 16k^2 \alpha, \\ C_2 = -\beta^2 + 8k^2 + 4\beta k^2 + 8k^2 \alpha + 2\beta^2 a_0^2, \\ C_3 = -4k^2 - 2\beta k^2 - 4k^2 \alpha + \beta^2 a_1^2, \\ C_4 = -\beta a_0 + 2k^2 a_0 + 2\beta a_0^3 - k c a_1. \end{cases}$$

此代数方程组中的 α, β 为参数, k, a_0, a_1, c 为变元. 我们通过对参数的调整可以求得此代数多项式中在不同参数值下变元的近似值. 譬如我们取 $\alpha=0, \beta=1$, 可以得到变元 k, a_0, a_1, c 的近似数值为 (改进的 Newton 迭代法求解代数方程组的程序和运行结果见附录 4):

$$k = 0.2041, a_0 = 0.5000, a_1 = -0.5000, c = 2.0412$$

我们将其代入 (5.4.3), 可得方程 (5.4.2) 的近似解为:

$$v(x, t) = 0.5000 - 0.5000 \tanh[0.2041(x - 2.0412)]$$

进而得到 (5.4.1) 中的近似解为:

$$u(x, t) = \{0.5000 - 0.5000 \tanh[0.2041(x - 2.0412)]\}^2.$$

5.5 结论

将符号方法和数值方法结合起来求解（偏）微分方程的方法可以解决很多难于求解的微分方程模型，本文所用到的方法对很多其他的微分方程模型也可以类似求解，在此不一一列举。混合方法求解微分方程的思想方法有着广泛的应用前景和研究价值，不过本文只是在这一方面作了一点初步的探索工作，但本文引导了我们求解微分方程的新思维和新视角，我们还可以尝试着将符号计算和数值计算的其他方法相结合来求解比较复杂的偏微分方程，这是一个新领域，这一研究领域值得我们继续去探索和研究，希望能有更多的相关研究成果出现。

参考文献

- [1]陆文端.微分方程中的变分方法.北京:科学出版社,2003.211-227
- [2]王东明,夏壁灿.计算机代数.北京:清华大学出版社,2004.113-129
- [3]吴文俊.数学机械化.北京:科学出版社,2003.114-120
- [4]柳银萍,李志斌.非线性微分方程一类精确解的计算机自动推导.华东师范大学学报.2003,1:1000-5641
- [5]王东明等.符号计算选讲.北京:清华大学出版社,2003.220-234
- [6]李志斌,张善卿.非线性波方程准确孤立波解的符号计算.数学物理学报.1997,17(1):81-89
- [7]Hunter J K, et al. Solitary and periodic gravity-waves of finite Amplitude . J. Fluid Mech, 1983, 56:3052-3054
- [8]张恭庆,林源渠.泛函分析讲义(上册).北京:北京大学出版社,1987.100-112
- [9]张恭庆等.泛函分析讲义(下册).北京:北京大学出版社,1990.240-248
- [10]Dai Xianxi, et al. Some solitary wave solutions for families of Generalized higher order KDV equations. Phys. Lett., A, 1989, 142:367-371
- [11]宋兆基等. MATLAB6.5 在科学计算中的应用.北京:清华大学出版社,2005.215-230
- [12]黎捷. MAPLE9.0 符号处理及应用.北京:科学出版社,2004.265-270
- [13]Lan Huibin, et al. Exact solutions for some nonlinear equations. Phy. Lett A, 1989, 137:369-372
- [14]Ben-Isreal. A modified Newton-Raphson method for the solution Of systems of equations. J. Math., 1965, 3:94-98
- [15]Ben-Isreal. A Newton-Raphson method for the solution of systems of equations. J. Math. Anal. Applic., 1966, 15:243-252
- [16]Tolga Guyer, seref Mirasyedioglu. A symbolic computation approach to a three-dimensional inverse problem for the transport equation. Applied Mathematic and computation, 2004, 154:181-193
- [17]YURI LEVIN AND ADI BEN-ISRAEL. Directional Newton method in n variables. Mathamatics of Computation Science , 2001, 237:251-262

- [18] Jean Moulin ollagnier. Algorithms and methods in differential algebra. Theoretical Computation Science, 1996, 157:115-127
- [19] Nikolaos I. Ioakimidis. A numerical replacement of computer Algebra methods for the derivation of polynomial equations in mechanics. Advance in Engineering Software, 1997, 28:539-547
- [20] Bernard P. Brooks. The coefficients of the characteristic polynomial in terms of the eigenvalues and the elements of an $n \times n$ matrix . Applied Mathematics letter, 2006, 19:511-555
- [21] Driss Bouziane, Abdelilah Kandri Rody and Hamid Maarouf . Unmixed-dimensional Decompositoin of a Finitely Generated Perfect Differrential Ideal. J. Symbolic Computation, 2001, 31:631-649
- [22] Rosemary Baines, Peter Vamos. An algorithm to compute the set of characteristics of polynomial equations over the integers. Journal of Symbolic Computation, 2003, 35:269-279
- [23] Moulay A. Barkatou. On Rational Solutions of Systems of Linear Differential Equations. J. Symbolic Computation, 1999, 28:547-567
- [24] Xiao-shan Gao . Implicitization of differential rational parametric equations. Journal of Symbolic Computation, 2003, 36:811-824
- [25] Eleni. nastasselou. Compatibility differential/difference equations in classical beam problems with the characteristic sets method. Advance in Engineering Software, 1995, 22:63-68
- [26] G. Baumann. Symmetry Analysis of Nonlinear Differential Equations with the "Mathematic" Program SYMMAN. Mathl. Comput., 1997, Modeling Vol. 25, No. 8/9, pp. :141-152
- [27] F. Faghihi, K. Hadad. Numerical solution of coupled differential equations and initial values using Maple software. Applied Mathematics and Computation, 2004, 155: 563-572
- [28] BRAM DE JAGER AND BRAM VAN ASCH. Symbolic Solutions for a class of Partial Differrential Equations. J. Symboilic Computation, 1996, 22: 459-468
- [29] Mustafa Bayram, Ercan Celik. Simultaneous solution of polynomial equations. Applied Mathematics and Computation, 2002, 133:533-538
- [30] Eisenstat, S., Walker, H. Choosing the forcing terms in an inexact Newton method. SIAM J. Sci. Comput., 1996, 17:16-32

- [31] Onur Kiyamaz, Seref Mirasyedioglu. A new symbolic computational approach to singular initial value problems in the second-order ordinary differential equations. *Applied Mathematics and Computation* 2005, 171:1218-1225
- [32] Isaacson, E., Keller, H. B. *Analysis of Numerical method*. New York, John Wiley and Sons. 1996
- [33] Ercan Celik, Mustafa Bayram. On the numerical solution of differential-algebraic equations by Pade series. *Applied Mathematics and Computation*, 2003, 137:151-160
- [34] Ercan Celik, Erdal Karaduman, Mustafa Bayram. Numerical solutions of chemical differential-algebraic equations. *Applied Mathematics and Computation*, 2003, 139:259-264
- [35] Ercan Celik. On the numerical solution of chemical differential-algebraic equations by Pade series. *Applied Mathematics and Computation*, 2004, 153:13-17
- [36] Ercan Celik, Mustafa Bayram. Arbitrary order numerical method for solving differential-algebraic equation by Pade series. *Applied Mathematics and Computation*, 2003, 137: 57-65
- [37] Nuran Guzel, Mustafa Bayram. On the numerical solution of stiff systems. *Applied Mathematics and Computation*, 2005, 170:230-236
- [38] Ercan Celik, Mustafa Bayram. Numerical solution of differential-algebraic equation systems and applications. *Applied Mathematics and Computation*, 2004, 154 : 405-413
- [39] Weiming Wang. A mechanical algorithm for solving the Volterra integral equation. *Applied Mathematics and Computation*, 2006, 172: 1323-1341
- [40] B. Hu, P. Eberhard and W. Schiehlen. Symbolical impact analysis for a falling conical rod against the rigid ground. *Journal of Sound and Vibration*, 2001, 240(1): 41-57
- [41] Abdul-Majid Wazwaz. The tanh method: exact solutions of the sine-Gordon and the sinh-Gordon equations. *Applied Mathematics and Computation*, 2005, 167: 1196-1210
- [42] A. -M. Wazwaz. A Sine-Cosine Method for Handling Nonlinear Wave Equations. *Mathematical and computer Modelling*, 2004, 40:499-508

- [43] Harry G. Kwatny, Bor-Chin Chang. Symbolic computing of nonlinear observer and forms. *Applied Mathematics and Computation* 2005, 171: 1058-1080
- [44] G. Baumann. Symmetry Analysis of Differential Equations with Mathematica. *Mathl Comput. Modelling*, 1997, Vol. 25, No. 8/9, pp: 25-37
- [45] M. C. Nucci. The Role of Symmetries in Solving Differential Equations. *Mathl. Comput. Modelling*, 1997, Vol. 25, No. 8/9, pp: 181-193
- [46] E. S. Cheb-Terrab, L. G. S. Duarte, L. A. C. P. da Mota. Computer algebra solving of first order ODEs using Symmetry methods. *Computer Physics Communications*, 1997, 101: 254-268
- [47] Serge Andrianov. Symbolic computation of approximate symmetries for ordinary differential equations. *Mathematics and Computers in Simulation*, 2001, 57: 147-153
- [48] E. S. Cheb-Terrab, L. G. S. Duarte, L. A. C. P. da Mota. Computer algebra solving of second order ODEs using Symmetry methods. *Computer Physics Communications*, 1998, 108: 90-114
- [49] Ruo-xia Yao, Zhi-bin Li. New exact solutions for three nonlinear evolution equations. *Physics Letters A*, 2002, 297: 1996-204
- [50] Bican Xia, Xiaorong Hou. A Complete Algorithm for Counting Real Solutions of Polynomial Systems of Equations and Inequalities. *Computers and Mathematics with Applications*, 2002, 44: 633-642
- [51] B. Bulos, Suheil A. Khuri. On the modified Taylor's approximation for the solution of linear and nonlinear equations. *Applied Mathematics and Computation*, 2005, 160: 939-953
- [52] G. Scheifele. On numerical integration of perturbed linear oscillating systems, *Z. Agnew. Math. phys.*, 1971, 22: 186-210
- [53] G. Adomian. A review of the decomposition method in applied mathematics. *J. Math. Anal. Appl.*, 1988, 135: 501-544
- [54] G. Adomian. A new approach to nonlinear partial differential Equations. *J. Math. Anal. Appl.*, 1984, 102: 420-434
- [55] G. Adomian. A review of the composition method and some recent results for nonlinear equations. *Math. Comput. Model*, 1992, 13(7): 17
- [56] G. Corliss, Y. F. Chang. Solving ordinary differential equations using Taylor series. *ACM Trans. Math. Soft*, 1982, 8: 114-144

- [57] P. Amodio, F. Mazzia. Numerical solution of differential-algebraic equations and computation of consistent initial/boundary conditions. *J. Comput. Appl. Math.*, 1997, 87:135-146
- [58] H. Hirayama. Arbitrary order and A-stable numerical method for solving algebraic ordinary differential equation by power series. In: *International Conference on Mathematics and Computers in Physics*, Vouliagmeni, Athens, Greece, 2000, 7:9-16
- [59] M. Bayram. Application of computer algebra techniques to affinity binding equations. *Appl. Math. Comput.*, 1998, 94:83-90
- [60] M. Bayram. Parameter estimation of the enzyme kinetic system using computer algebra. *Appl. Math. Comput.*, 1999, 99(1):98-99
- [61] 吕克璞, 石玉仁等. KdV-Burgers 方程的孤立波解. *物理学报*, 2001, 50(11):1000-3290
- [62] 张善卿, 李志斌. Jacobi 椭圆函数展开法的新应用. *物理学报*, 2003, 52(05):1000-3290
- [63] 王晓丽, 张鸿庆. Jacobi 椭圆函数展开法及其应用. *烟台大学学报*, 2003, 16(2):1004-8820
- [64] 范恩贵, 张鸿庆. 非线性孤子方程的齐次平衡法. *物理学报*, 1998, 47:353-362
- [65] 闫振亚, 张鸿庆. 非线性浅水长波近似方程组的显示精确解. *物理学报*, 1999, 48:1962-1968
- [66] Zhangshanqing, LiZhibin. A Package for Generating the Determining System of Nonclassical symmetry of Differential Equations. *Numer. Math.*, 2005, 1:1000-5641
- [67] 张桂戎, 李志斌, 段一士. 非线性波方程精确孤立波解. *中国科学 (A 辑)* 30(12)
- [68] 殷志云. 微分方程及其中心流形方法的自动化简. *数学的实践与认识*, 2003, 33(6)
- [69] P. Deufhard, E. Hairer, J. Zugck. One step and extrapolation methods for differential-algebraic systems. *Numer. Math.*, 1987, 51:501-516
- [70] B. Leimkuhler L. R. Petzold, C. W. Gear. Approximation methods for the consistent initialization of differential equations. *SIAM J. Numer. Anal.*, 1991, 28:205-226
- [71] Atkinson, K. E. *An Introduction to Numerical Analysis*. New york,

John Wiley and Sons. 1978

[72] Ben-Israel, A., Greville, T. N. E. Generalized Inverse: Theory and Applications. New York, John Wiley and Sons. 1974

[73] Boullion, T. L., Odell, P. L. Generalized Inverse Matrices. New York, Wiley-Interscience. 1971

[74] Dembo, R. S., Eisenstat, S. C., Steihaug, T. Inexact Newton method. SIAM J. Numer. Anal., 1982, 19:400-408

附 录

附录 1: Adomian 分解方法的计算程序.

```

ADM:=proc(h, f, g, a, b)
global function_y; local i, poly_A, sum;
  function_y[0]:=a+b*x+inverse_L(h, g);
  poly_A[0]:=subs(y=function_y[0], f);
  sum:=function_y[0];
  for i from 1 to 4 do
    function_y[i]:=-inverse_L(h, poly_A[i-1]);
    sum:=sum+function_y[i];
    poly_A[i]:=A_Poly(f, i); od;
  RETURN(y(x)=series(sum, x, 10));
end:
inverse_L:=proc(h, g)
local Inner_Int, Outer_Int;
  Inner_Int:=int(h*g, x=0..t);
  Outer_Int:=int(subs(x=t, 1/h)*Inner_Int, t=0..x);
RETURN(Outer_Int);
end:
A_Poly:=proc(f, n)
local term;
  if n=1 then term:=function_y[1]*subs(y=function_y[0], diff(f, y)) fi;
  if n=2 then
term:=function_y[2]*subs(y=function_y[0], diff(f, y))+function_y[1]^2*
  subs(y=function_y[0], diff(f, y$2)/2!) fi;
  if n=3 then
term :=function_y[3]*subs(y=function_y[0], diff(f, y))+function_y[1]*
  function_y[2]*subs(y=function_y[0], diff(f, y$2))+function_y[1]^3*
  subs(y=function_y[0], diff(f, y$3))/3! fi;
  if n=4 then term:=function_y[4]*subs(y=function_y[0], diff(f, y))+
  (function_y[1]*function_y[3]+function_y[2]^2/2!)*

```

```
subs(y=function_y[0], diff(f, y$2))+function_y[1]^2*function_y[2]/2!*  
    subs(y=function_y[0], diff(f, y$3))+1/4*function_y[1]^4*  
    subs(y=function_y[0], diff(f, y$4)) fi;  
    RETURN(term);  
end
```

附录 2: 在数学机械化“自动推理平台”环境中, 我们计算特征列的实例演示.

```
>charset([a0-c, 2*m*k+a1], [a0, a1]);
```

```
[a0-c, a1+2*k*m]
```

```
>charset([(2*k^2*m+c-a0)*a1, a1^2+16*k^2*m*a2+2*c*a2-2*a0*a2, a1*(8*k^2*m+c-a0+3*a2), -a1^2+40*k^2*m*a2+2*c*a2-2*a0*a2+2*a2^2, a1*(2*k^2*m+a2), 12*k^2*m+a2], [a0, a1, a2]);
```

```
[-240*k^4*m^2*a0+240*k^4*m^2*c+1920*k^6*m^3, -10*k^2*m*a1, 12*k^2*m+a2]
```


附录 3: 将微分代数方程的幂级数解展开到任意次的程序及运行结果.

```
function Series()

syms e1 e2 x;
n=10;
digits(10);
e=[e1;e2];
A=[1,-x;0,0];
B=[1,-(1+x);0,1];
g=[0;sin(x)];
a(1)=1;a(2)=-1;
b(1)=0;b(2)=1;
v=[a(1)+a(2)*x;b(1)+b(2)*x];
h=x^2;
for i=2:n
    v=v+e*h;
    dv=diff(v,x);
    H=A*dv+B*v-g;
    H=[diff(H(1),x,i-1);diff(H(2),x,i)];
    H=subs(H,x,0);
    gg=-subs(H,e,[0;0]);
    AA=jacobian(H,e);
    dd=(AA^-1)*gg;
    a(i+1)=dd(1);
    b(i+1)=dd(2);
    h=h*x;
    v(1)=subs(v(1),e1,a(i+1));
    v(2)=subs(v(2),e2,b(i+1));
end
vpa(v(1))
vpa(v(2))

ans =
```

```
1.-1.*x+1.500000000*x^2-.1666666667*x^3-.1250000000*x^4-.8333333333e-  
2*x^5+.9722222222e-2*x^6-.1984126984e-3*x^7-.1736111111e-3*x^8-.27557  
31922e-5*x^9+.3031305115e-5*x^10
```

```
ans =
```

```
x-.1666666667*x^3+.8333333333e-2*x^5-.1984126984e-3*x^7+.2755731922e-  
5*x^9
```

附录 4: 改进的 Newton 迭代法求解代数方程组的程序和运行结果.

```

function y=Newton()
ERR=0.001;err=0;
beta=1;
afa=0;
syms k a0 a1 c;
x=[k;a0;a1;c];
ix=[0.15;0.8;-0.6;2.1];
nx=ix;
C1=-beta^2+16*k^2+8*beta*k^2+16*k^2*afa;
C2=-beta^2+8*k^2+4*beta*k^2+8*k^2*afa+2*beta^2*a0^2;
C3=-4*k^2-2*beta*k^2-4*k^2*afa+beta^2*a1^2;
C4=-beta*a0+2*k^2*a0+2*beta*a0^3-k*c*a1;
F=[C1;C2;C3;C4];
J=jacobian(F,x);

for i=1:10
    Jx=double(subs(J,x,ix));
    Fx=double(subs(F,x,ix));
    nx=ix-Jx'*(Jx*Jx')^(-1)*Fx;
    err=sum(abs(nx-ix));
    if err<ERR
        break;
    end
    ix=nx;
end
y=nx;

```

To get started, type one of these commands: helpwin, helpdesk, or demo

?Newton

nx =

0.2139

0.5563

-0.5083

4.0103

nx =

0.2043

0.5028

-0.5001

2.2335

nx =

0.2041

0.5000

-0.5000

2.0417

nx =

0.2041

0.5000

-0.5000

2.0412

ans =

0.2041

0.5000

-0.5000

2.0412

致 谢

衷心感谢恩师殷志云教授!殷老师病重期间仍一直在督促指导我的学习和论文工作,殷老师的人格魅力深深地影响了我。殷老师深厚的数学功底,渊博的知识背景,严谨的治学态度,一丝不苟的工作作风让我深受启迪。近三年来殷老师在我的学业上倾注了大量的心血,我所取得的每一点进步都饱含了殷老师无私的付出。本文亦是在殷老师精心指导下完成的,本论文无论是在选题、推导过程还是完稿后对文字部分的不断完善,都得到了殷老师的细心指导。在我研究生学习期间,殷老师还对我的生活关怀备致,在此我对殷老师表示诚挚的谢意!

感谢中南大学数学院在我作论文期间为我提供了优良的学习环境,使得我的论文得以顺利完成。感谢陈小松教授,唐先华教授为我的论文审稿付出了辛勤的劳动。感谢中南大学数学院的全体老师和同学,正是你们营造的良好学术氛围和学习环境让我深受启发,受益非浅。

在此,我还要特别感谢王伟同学,他在我论文期间,给予了我极大的帮助。感谢我的室友王志同,蔡白光,黄介武三年以来,给予我物质和学习上的帮助。感谢所有关心过我的老师和陪伴我度过三年美好时光的同学们,衷心的祝福你们生活幸福美满!

最后,我要特别感谢我的亲人和我的朋友,正是你们在物质和精神上给我的支持和理解,让我渡过了一个又一个难关,顺利完成学业,在这里我向你们表达我深深的谢意!

蒋青松于中南大学铁道学院

2006年10月

攻读学位期间的主要研究成果

- [1] 蒋青松, 蒋茂旺. 牛顿迭代法的改进及其在 BANACH 空间中的收敛性, 数学理论与应用. (已接受) 拟定于 2006 年第 4 期发表.
- [2] 孙伟, 蒋青松. 对修理期间部件无老化的串联系统可用度的进一步探讨, 数学的实践与认识. 2006 年第 8 期.
- [3] 符号—数值混合方法求解微分方程的方法探索及其机器实现. (已投稿)
- [4] 将微分代数方程的解展开到任意次的机器实现方法. (已投稿)